

数据治理中心

# API 参考

文档版本 01  
发布日期 2025-01-22



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

---

# 目录

---

<b>1 使用前必读</b>	<b>1</b>
1.1 概述	1
1.2 调用说明	1
1.3 基本概念	1
1.4 终端节点 Endpoint	2
1.5 项目 ID 和账号 ID	4
1.6 实例 ID 和工作空间 ID	5
1.7 数据资产 guid	5
1.8 约束与限制	7
<b>2 API 概览</b>	<b>9</b>
2.1 数据集成 API 概览	9
2.2 数据开发 API (V1) 概览	12
2.3 数据开发 API (V2) 概览	14
2.4 管理中心 API 概览	16
2.5 数据架构 API 概览	18
2.6 数据质量 API 概览	32
2.7 数据目录 API 概览	35
2.8 数据服务 API 概览	40
2.9 数据安全 API 概览	45
<b>3 如何调用 API</b>	<b>53</b>
3.1 构造请求	53
3.2 认证鉴权	57
3.3 返回结果	59
<b>4 数据集成 API</b>	<b>61</b>
4.1 集群管理	61
4.1.1 查询集群详情	61
4.1.2 删除集群	71
4.1.3 查询所有可用区	75
4.1.4 查询支持的版本	80
4.1.5 查询版本规格	84
4.1.6 查询规格详情	89
4.1.7 查询所有集群的企业项目 ID	93

4.1.8 查询集群的企业项目 ID.....	98
4.1.9 查询集群实例信息.....	100
4.1.10 修改集群.....	106
4.1.11 重启集群.....	111
4.1.12 启动集群.....	116
4.1.13 停止集群（待下线）.....	120
4.1.14 创建集群.....	122
4.1.15 查询集群列表.....	132
4.2 作业管理.....	141
4.2.1 查询作业.....	141
4.2.2 删除作业.....	152
4.2.3 修改作业.....	154
4.2.4 随机集群创建作业并执行.....	172
4.2.5 停止作业.....	190
4.2.6 指定集群创建作业.....	194
4.2.7 启动作业.....	212
4.2.8 查询作业状态.....	214
4.2.9 查询作业执行历史.....	220
4.3 连接管理.....	226
4.3.1 创建连接.....	226
4.3.2 查询连接.....	238
4.3.3 删除连接.....	245
4.3.4 修改连接.....	249
4.4 附：公共数据结构.....	261
4.4.1 连接参数说明.....	261
4.4.1.1 关系数据库连接.....	261
4.4.1.2 OBS 连接.....	264
4.4.1.3 HDFS 连接.....	266
4.4.1.4 HBase 连接.....	268
4.4.1.5 CloudTable 连接.....	271
4.4.1.6 Hive 连接.....	273
4.4.1.7 FTP/SFTP 连接.....	275
4.4.1.8 MongoDB 连接.....	276
4.4.1.9 Redis 连接.....	277
4.4.1.10 Kafka 连接.....	278
4.4.1.11 DIS 连接.....	280
4.4.1.12 Elasticsearch/云搜索服务(CSS)连接.....	281
4.4.1.13 DLI 连接.....	282
4.4.1.14 DMS Kafka 连接.....	283
4.4.2 源端作业参数说明.....	284
4.4.2.1 源端为关系数据库.....	284
4.4.2.2 源端为对象存储.....	286

4.4.2.3 源端为 HDFS.....	291
4.4.2.4 源端为 Hive.....	296
4.4.2.5 源端为 HBase/CloudTable.....	297
4.4.2.6 源端为 FTP/SFTP.....	299
4.4.2.7 源端为 HTTP/HTTPS.....	303
4.4.2.8 源端为 MongoDB/DDS.....	304
4.4.2.9 源端为 Redis.....	306
4.4.2.10 源端为 DIS.....	307
4.4.2.11 源端为 Kafka.....	309
4.4.2.12 源端为 Elasticsearch/云搜索服务.....	310
4.4.3 目的端作业参数说明.....	311
4.4.3.1 目的端为关系数据库.....	311
4.4.3.2 目的端为 OBS.....	314
4.4.3.3 目的端为 HDFS.....	318
4.4.3.4 目的端为 Hive.....	321
4.4.3.5 目的端为 HBase/CloudTable.....	322
4.4.3.6 目的端为 DDS.....	324
4.4.3.7 目的端为 Elasticsearch/云搜索服务.....	325
4.4.3.8 目的端为 DLI.....	327
4.4.3.9 目的端为 DIS.....	328
4.4.4 作业任务参数说明.....	329
<b>5 数据开发 API ( V1 ) .....</b>	<b>333</b>
5.1 脚本开发 API.....	333
5.1.1 创建脚本.....	333
5.1.2 修改脚本内容.....	337
5.1.3 查询脚本信息.....	340
5.1.4 查询脚本列表.....	343
5.1.5 查询脚本实例执行结果.....	347
5.1.6 删除脚本.....	350
5.1.7 执行脚本.....	351
5.1.8 停止脚本实例的执行.....	353
5.2 资源管理 API.....	355
5.2.1 创建资源.....	355
5.2.2 修改资源.....	357
5.2.3 查询资源详情.....	359
5.2.4 删除资源.....	361
5.2.5 查询资源列表.....	363
5.3 作业开发 API.....	365
5.3.1 创建作业.....	365
5.3.2 修改作业.....	387
5.3.3 查询作业列表.....	408
5.3.4 查询作业详情.....	411

5.3.5 查询作业文件.....	432
5.3.6 导出作业.....	434
5.3.7 批量导出作业.....	436
5.3.8 导入作业.....	438
5.3.9 立即执行作业.....	441
5.3.10 启动作业.....	443
5.3.11 停止作业.....	445
5.3.12 删除作业.....	446
5.3.13 停止作业实例.....	448
5.3.14 重跑作业实例.....	449
5.3.15 查询实时作业运行状态.....	451
5.3.16 查询作业实例列表.....	454
5.3.17 查询作业实例详情.....	458
5.3.18 查询系统任务详情.....	462
5.4 连接管理 API (待下线) .....	465
5.4.1 创建连接 (待下线) .....	465
5.4.2 查询连接列表 (待下线) .....	472
5.4.3 查询连接详情 (待下线) .....	474
5.4.4 修改连接 (待下线) .....	476
5.4.5 删除连接 (待下线) .....	479
5.4.6 导出连接 (待下线) .....	480
5.4.7 导入连接 (待下线) .....	481
<b>6 数据开发 API (V2) .....</b>	<b>485</b>
6.1 作业开发 API.....	485
6.1.1 创建补数据实例.....	485
6.1.2 查询补数据实例.....	488
6.1.3 停止补数据实例.....	492
6.1.4 修改作业名称.....	493
6.1.5 查询发布包列表.....	495
6.1.6 查询指定发布包详情.....	500
6.1.7 设置作业标签.....	504
6.1.8 查询告警通知记录.....	506
6.1.9 发布任务包.....	509
6.1.10 撤销任务包.....	511
6.1.11 查询实例运行状态.....	513
6.1.12 查询任务完成情况.....	515
6.1.13 查询指定作业的实例列表.....	518
6.1.14 重跑作业实例.....	522
<b>7 管理中心 API.....</b>	<b>526</b>
7.1 数据连接管理.....	526
7.1.1 查询数据连接列表.....	526
7.1.2 创建数据连接.....	531

7.1.3 测试创建数据连接.....	536
7.1.4 查询单个数据连接信息.....	541
7.1.5 更新数据连接信息.....	546
7.1.6 删除数据连接.....	551
7.2 购买实例.....	555
7.2.1 DataArtsStudio 实例一键购买接口.....	556
7.3 工作空间管理.....	561
7.3.1 获取工作空间列表.....	561
7.3.2 创建工作空间.....	566
7.3.3 获取单个工作空间信息.....	571
7.4 实例管理.....	576
7.4.1 获取实例列表.....	576
7.5 工作空间用户管理.....	581
7.5.1 获取工作空间用户角色.....	581
7.5.2 编辑工作空间用户或用户组.....	586
7.5.3 获取工作空间用户信息.....	592
7.5.4 添加工作空间用户.....	597
7.5.5 删除工作空间用户.....	602
7.6 数据源元数据获取.....	607
7.6.1 获取数据库列表.....	607
7.6.2 获取 schemas.....	612
7.6.3 获取数据源中的表.....	617
7.6.4 获取数据源中表的字段.....	623
7.7 实例规格变更.....	628
7.7.1 规格变更接口.....	628
<b>8 数据架构 API.....</b>	<b>634</b>
8.1 概览.....	634
8.1.1 总览统计信息.....	634
8.1.2 模型统计信息.....	641
8.1.3 关系建模统计信息.....	659
8.1.4 标准覆盖率统计信息.....	673
8.2 信息架构接口.....	690
8.2.1 查询多种类型的表信息.....	690
8.3 数据标准接口.....	716
8.3.1 获取数据标准集合.....	716
8.3.2 创建数据标准.....	732
8.3.3 删除数据标准.....	760
8.3.4 修改数据标准.....	767
8.3.5 查看数据标准详情.....	796
8.3.6 关联属性与数据标准.....	810
8.4 数据源接口.....	818
8.4.1 获取数据连接信息.....	818

8.5 码表管理接口.....	824
8.5.1 查询码表列表.....	824
8.5.2 创建码表.....	839
8.5.3 删除码表.....	864
8.5.4 修改码表.....	870
8.5.5 查看码表详情.....	897
8.5.6 查看码表字段值.....	910
8.5.7 编辑码表字段值.....	917
8.6 流程架构接口.....	932
8.6.1 获取所有流程架构目录树.....	932
8.6.2 查询流程架构列表.....	939
8.6.3 创建流程架构.....	947
8.6.4 修改流程架构.....	955
8.6.5 删除流程架构.....	963
8.6.6 查找流程架构详情.....	969
8.7 数据标准模板接口.....	976
8.7.1 查询数据标准模板.....	976
8.7.2 创建数据标准模板.....	990
8.7.3 修改数据标准模板.....	999
8.7.4 删除数据标准模板.....	1008
8.7.5 初始化数据标准模板.....	1014
8.8 审批管理接口.....	1035
8.8.1 获取审批单.....	1036
8.8.2 撤回审批单.....	1049
8.8.3 审批单处理.....	1055
8.8.4 批量发布.....	1063
8.8.5 批量下线.....	1074
8.8.6 创建审批人.....	1086
8.8.7 查询审批人列表.....	1094
8.8.8 删除审批人.....	1101
8.8.9 删除实体下展.....	1107
8.8.10 获取下展信息与已发布实体的差异.....	1112
8.9 主题管理接口.....	1130
8.9.1 查找主题列表.....	1130
8.9.2 删除主题.....	1141
8.9.3 创建主题.....	1147
8.9.4 修改主题.....	1161
8.9.5 获取主题树信息.....	1175
8.9.6 查找主题列表(新).....	1214
8.9.7 删除主题(新).....	1225
8.9.8 创建主题(新).....	1232
8.9.9 修改主题(新).....	1244



8.10 主题层级接口.....	1256
8.10.1 获取主题层级.....	1256
8.10.2 修改或删除主题层级.....	1263
8.11 目录管理.....	1273
8.11.1 获取所有目录.....	1273
8.11.2 创建目录.....	1280
8.11.3 修改目录.....	1287
8.11.4 删除目录.....	1295
8.12 原子指标接口.....	1300
8.12.1 查找原子指标.....	1300
8.12.2 新建原子指标.....	1317
8.12.3 更新原子指标.....	1344
8.12.4 删除原子指标.....	1370
8.12.5 查看原子指标详情.....	1376
8.13 衍生指标接口.....	1391
8.13.1 查找衍生指标.....	1391
8.13.2 新建衍生指标.....	1422
8.13.3 更新衍生指标.....	1444
8.13.4 删除衍生指标.....	1489
8.13.5 查看衍生指标详情.....	1495
8.14 复合指标接口.....	1525
8.14.1 查找复合指标.....	1525
8.14.2 新建复合指标.....	1543
8.14.3 更新复合指标.....	1572
8.14.4 删除复合指标.....	1601
8.14.5 查看复合指标详情.....	1607
8.15 维度接口.....	1624
8.15.1 查找维度.....	1624
8.15.2 新建维度.....	1654
8.15.3 更新维度.....	1703
8.15.4 删除维度.....	1750
8.15.5 查看维度详情.....	1756
8.15.6 查看维度颗粒度.....	1787
8.15.7 查看逆向维度表任务.....	1847
8.16 限定接口.....	1853
8.16.1 查找业务限定.....	1853
8.16.2 查看限定详情.....	1877
8.17 维度表接口.....	1890
8.17.1 查找维度表.....	1890
8.17.2 删除维度表.....	1919
8.17.3 查看维度表详情.....	1923
8.18 事实表接口.....	1950

8.18.1 查找事实表.....	1950
8.18.2 删除事实表.....	1997
8.18.3 查看事实表详情.....	2003
8.18.4 查看逆向事实表任务.....	2048
8.19 汇总表接口.....	2054
8.19.1 查找汇总表.....	2054
8.19.2 新建汇总表.....	2084
8.19.3 更新汇总表.....	2130
8.19.4 删除汇总表.....	2155
8.19.5 查看汇总表详情.....	2161
8.20 业务指标接口.....	2190
8.20.1 查询业务指标信息.....	2190
8.20.2 创建业务指标.....	2212
8.20.3 更新业务指标.....	2246
8.20.4 删除业务指标.....	2280
8.20.5 查看指标详情.....	2286
8.20.6 查看指标维度信息.....	2305
8.20.7 查看指标责任人信息.....	2311
8.20.8 获取指标关联信息.....	2316
8.21 版本信息接口.....	2322
8.21.1 查找版本信息.....	2322
8.21.2 比较版本信息.....	2335
8.22 关系建模接口.....	2348
8.22.1 查找表模型列表.....	2348
8.22.2 创建表模型.....	2383
8.22.3 更新表模型.....	2435
8.22.4 删除表模型.....	2494
8.22.5 查询关系.....	2500
8.22.6 查看关系详情.....	2509
8.22.7 查询模型下所有关系.....	2516
8.22.8 查看表模型详情.....	2550
8.22.9 获取模型.....	2584
8.22.10 新建模型工作区.....	2591
8.22.11 更新模型工作区.....	2600
8.22.12 删除模型工作区.....	2608
8.22.13 查询模型详情.....	2614
8.22.14 查询目的表和字段(待下线).....	2620
8.22.15 导出模型中表的 DDL 语句.....	2626
8.22.16 转换逻辑模型为物理模型.....	2631
8.22.17 获取操作结果.....	2640
8.23 导入导出接口.....	2647
8.23.1 导入模型, 关系建模, 维度建模, 码表, 业务指标以及流程架构.....	2647

8.23.2 导入主题.....	2653
8.23.3 查询导入结果.....	2658
8.23.4 导出业务数据.....	2664
8.23.5 获取 excel 导出结果.....	2671
8.24 自定义项接口.....	2677
8.24.1 查询自定义项.....	2677
8.24.2 修改自定义项.....	2684
8.25 标签接口.....	2693
8.25.1 添加标签.....	2693
8.25.2 删除标签.....	2699
8.26 质量规则接口.....	2704
8.26.1 更新表的异常数据输出配置.....	2704
8.26.2 清空质量规则.....	2740
<b>9 数据质量 API.....</b>	<b>2747</b>
9.1 目录接口.....	2747
9.1.1 获取作业目录.....	2747
9.2 规则模板接口.....	2752
9.2.1 获取规则模板列表.....	2752
9.2.2 创建规则模板.....	2757
9.2.3 获取规则模板详情.....	2764
9.2.4 更新规则模板.....	2769
9.2.5 批量删除规则模板.....	2776
9.3 质量作业接口.....	2780
9.3.1 获取质量作业列表 V1.....	2780
9.3.2 获取质量作业列表.....	2785
9.3.3 获取质量作业详情.....	2791
9.3.4 批量删除质量作业.....	2796
9.4 对账作业接口.....	2798
9.4.1 获取对账作业列表.....	2799
9.4.2 获取对账作业详情.....	2804
9.4.3 批量删除对账作业.....	2811
9.5 运维管理接口.....	2813
9.5.1 获取任务执行结果列表.....	2813
9.5.2 获取实例结果.....	2819
9.5.3 数据质量运维管理操作处理&记录.....	2823
9.6 任务实例接口.....	2825
9.6.1 批量停止实例.....	2825
9.7 质量报告接口.....	2828
9.7.1 获取质量报告评分体系.....	2828
9.7.2 获取质量报告概览.....	2830
9.7.3 获取质量报告趋势.....	2832
9.7.4 获取质量报告规则评分.....	2835

9.7.5 获取质量报告子规则字段评分.....	2839
9.7.6 获取质量报告技术报告数据.....	2842
9.7.7 获取质量报告业务报告数据.....	2845
9.8 导入导出接口.....	2848
9.8.1 导出资源.....	2848
9.8.2 获取导入导出的任务状态.....	2850
9.8.3 下载资源任务文件.....	2852
9.8.4 上传导入的文件接口.....	2854
9.8.5 导入资源.....	2857
<b>10 数据目录 API.....</b>	<b>2860</b>
10.1 业务资产接口.....	2860
10.1.1 查询业务资产.....	2860
10.1.2 查询业务资产目录树.....	2868
10.2 指标资产接口.....	2874
10.2.1 查询指标资产目录树.....	2874
10.2.2 查询指标资产.....	2878
10.3 统计资产接口.....	2884
10.3.1 获取技术资产统计信息.....	2885
10.3.2 获取业务资产统计信息.....	2893
10.4 资产管理接口.....	2899
10.4.1 查询技术资产.....	2899
10.4.2 根据 guid 获取资产详情.....	2907
10.4.3 添加或修改资产.....	2913
10.4.4 删除资产.....	2921
10.5 资产分类接口.....	2926
10.5.1 批量资产关联分类.....	2926
10.5.2 资产关联分类.....	2931
10.5.3 移除资产关联的分类.....	2936
10.6 资产分级接口.....	2938
10.6.1 批量资产关联密级.....	2938
10.6.2 资产关联密级.....	2943
10.6.3 移除资产关联密级.....	2948
10.7 资产信息.....	2952
10.7.1 查询概要.....	2952
10.7.2 指定字段采集概要.....	2958
10.8 血缘信息.....	2964
10.8.1 无血缘关系表查询.....	2965
10.8.2 血缘查询.....	2974
10.8.3 血缘导入.....	2980
10.8.4 创建血缘信息.....	2987
10.9 元数据采集任务接口.....	2994
10.9.1 查询采集任务列表.....	2994

10.9.2 创建采集任务.....	2999
10.9.3 查询采集任务详情.....	3006
10.9.4 编辑采集任务.....	3012
10.9.5 删除单个采集任务.....	3018
10.9.6 启动、调度、停止采集任务.....	3023
10.9.7 获取任务日志.....	3027
10.10 数据地图接口.....	3032
10.10.1 元数据实时同步接口(邀测).....	3033
10.10.2 搜索查询标签分页展示(邀测).....	3040
10.10.3 用户行为分析(邀测).....	3046
10.10.4 资产搜索(邀测).....	3052
10.10.5 创建或修改资产(邀测).....	3062
10.10.6 资产详情(邀测).....	3069
10.10.7 资产血缘(邀测).....	3077
10.10.8 批量血缘(邀测).....	3088
10.10.9 表关联作业算子列表(邀测).....	3094
10.10.10 产出信息(邀测).....	3099
10.10.11 批量打标签(邀测).....	3105
10.11 标签接口.....	3109
10.11.1 查询标签列表.....	3110
10.11.2 标签关联到资产.....	3115
<b>11 数据服务 API.....</b>	<b>3120</b>
11.1 API 管理接口.....	3120
11.1.1 创建 API.....	3120
11.1.2 查询 API 列表.....	3151
11.1.3 更新 API.....	3159
11.1.4 查询 API 信息.....	3174
11.1.5 批量删除 API.....	3187
11.1.6 发布 API.....	3191
11.1.7 API 操作(下线/停用/恢复).....	3196
11.1.8 批量授权 API(专享版).....	3201
11.1.9 调试 API.....	3206
11.1.10 API 授权操作(授权/取消授权/申请/续约).....	3212
11.1.11 查看 API 发布信息(专享版).....	3215
11.1.12 查看 API 不同操作对应的实例信息(专享版).....	3220
11.1.13 查看 API 调试信息(专享版).....	3226
11.1.14 导入包含 API 信息的 EXCEL 文件.....	3231
11.1.15 导出包含 API 信息的 EXCEL 文件.....	3235
11.1.16 全量导出包含 API 的 EXCEL 压缩文件.....	3240
11.1.17 下载 EXCEL 模板.....	3246
11.2 申请管理接口.....	3249
11.2.1 查询申请列表.....	3250

11.2.2 审核申请.....	3256
11.2.3 获取申请详情.....	3260
11.3 消息管理接口.....	3263
11.3.1 查询消息列表.....	3263
11.3.2 处理消息.....	3269
11.3.3 获取消息详情.....	3273
11.4 授权管理接口.....	3279
11.4.1 查询 API 已授权的 APP.....	3279
11.4.2 查询 APP 已拥有授权的 API.....	3284
11.5 服务目录管理接口.....	3289
11.5.1 获取当前目录下的所有类型列表.....	3289
11.5.2 获取当前目录下的 API 列表.....	3294
11.5.3 获取当前目录下的目录列表（全量）.....	3301
11.5.4 更新服务目录.....	3306
11.5.5 查询服务目录.....	3311
11.5.6 创建服务目录.....	3315
11.5.7 批量删除目录.....	3320
11.5.8 移动当前目录至新目录.....	3325
11.5.9 批量移动 API 至新目录.....	3329
11.5.10 通过路径获取 ID.....	3333
11.5.11 通过 ID 获取路径.....	3337
11.5.12 通过目录 ID 获取路径对象.....	3342
11.5.13 查询服务目录 API 列表.....	3347
11.6 网关管理接口.....	3354
11.6.1 获取网关实例(专享版).....	3354
11.6.2 获取网关分组.....	3358
11.7 应用管理接口.....	3363
11.7.1 创建应用.....	3363
11.7.2 查询应用列表.....	3369
11.7.3 更新应用.....	3374
11.7.4 删除应用.....	3379
11.7.5 查询应用详情.....	3383
11.8 总览接口.....	3388
11.8.1 查询统计用户相关的总览开发指标.....	3388
11.8.2 查询统计用户相关的总览调用指标.....	3393
11.8.3 查询 API 服务调用 TopN.....	3395
11.8.4 查询 APP 服务使用 TopN.....	3401
11.8.5 查询 API 统计数据详情.....	3407
11.8.6 查询 APP 统计数据详情.....	3412
11.8.7 查询 API 仪表盘数据详情.....	3417
11.8.8 查询指定 API 仪表盘数据详情.....	3422
11.8.9 查询 APP 仪表盘数据详情.....	3427

11.8.10 查询指定 API 应用调用 TopN.....	3433
11.9 集群管理接口.....	3438
11.9.1 查询集群概览信息列表.....	3438
11.9.2 查询集群详情信息列表.....	3444
11.9.3 查询集群详情信息.....	3451
11.9.4 查询数据服务集群访问日志列表.....	3458
11.9.5 开启数据服务集群 OBS 日志转储.....	3461
11.9.6 开启数据服务集群 LTS 日志转储.....	3463
<b>12 数据安全 API.....</b>	<b>3466</b>
12.1 权限管理接口.....	3466
12.1.1 创建权限集.....	3466
12.1.2 查询权限集列表.....	3472
12.1.3 查询权限集.....	3478
12.1.4 删除权限集.....	3483
12.1.5 更新权限集.....	3486
12.1.6 添加权限集成员.....	3492
12.1.7 查询权限集成员列表.....	3497
12.1.8 批量删除权限集成员.....	3503
12.1.9 添加权限集的权限.....	3507
12.1.10 查询权限集的权限列表.....	3514
12.1.11 删除权限集的权限.....	3520
12.1.12 更新权限集的权限.....	3524
12.1.13 查询数据源可配置权限.....	3530
12.1.14 查询权限集中配置的 url 信息.....	3534
12.1.15 查询数据操作信息.....	3538
12.2 识别规则接口.....	3543
12.2.1 查询识别规则列表.....	3543
12.2.2 创建识别规则.....	3549
12.2.3 查询特定识别规则.....	3558
12.2.4 删除识别规则.....	3562
12.2.5 修改识别规则接口.....	3566
12.2.6 批量删除识别规则接口.....	3572
12.2.7 修改识别规则状态接口.....	3576
12.2.8 创建组合识别规则.....	3581
12.2.9 修改组合识别规则.....	3584
12.2.10 组合识别规则测试.....	3588
12.3 规则分组接口.....	3591
12.3.1 查询规则组列表.....	3591
12.3.2 创建规则分组接口.....	3597
12.3.3 查询规则组.....	3601
12.3.4 修改规则分组接口.....	3606
12.3.5 删除规则分组接口.....	3610

12.4 数据权限查询接口.....	3611
12.4.1 查询角色对一组库、表的可配置操作权限信息.....	3611
12.5 数据密级接口.....	3616
12.5.1 获取数据密级.....	3616
12.5.2 创建数据密级.....	3621
12.5.3 根据指定的 id 查询数据密级.....	3626
12.5.4 删除指定的 id 的数据密级.....	3630
12.5.5 根据指定的 id 修改数据密级.....	3634
12.5.6 批量删除数据密级.....	3639
12.6 权限应用接口.....	3642
12.6.1 查询数据开发细粒度连接列表.....	3643
12.6.2 批量更新数据开发连接细粒度认证状态.....	3647
12.6.3 测试数据开发连接细粒度连通性.....	3653
12.7 敏感数据分布接口.....	3656
12.7.1 查询敏感数据发现概览结果(以分类和密级为单位).....	3656
12.8 用户同步接口.....	3662
12.8.1 查询用户同步列表.....	3662
12.8.2 查询单个用户同步任务.....	3669
12.9 队列权限接口.....	3674
12.9.1 查询当前空间下分配的队列资源.....	3674
12.9.2 分配队列资源给指定空间.....	3679
12.9.3 修改当前空间下分配的队列资源.....	3685
12.9.4 删除当前空间下分配的队列资源.....	3690
12.10 数据分类接口.....	3693
12.10.1 导入预置分类.....	3693
12.10.2 查询数据分类列表.....	3698
12.11 数据安全诊断接口.....	3702
12.11.1 执行数据安全诊断.....	3702
12.11.2 查询不合理的权限配置.....	3703
12.11.3 查询数据权限控制模块诊断结果.....	3706
12.11.4 查询敏感数据保护模块诊断结果.....	3709
12.11.5 查询未进行静态脱敏任务的表信息.....	3712
12.11.6 查询数据源防护模块诊断结果.....	3715
12.12 空间资源权限策略管理接口.....	3718
12.12.1 查询空间资源权限策略列表.....	3718
12.12.2 创建空间资源权限策略.....	3722
12.12.3 查询单个资源权限策略.....	3726
12.12.4 更新空间资源权限策略.....	3728
12.12.5 批量删除资源权限策略.....	3732
12.13 安全管理员接口.....	3734
12.13.1 查看安全管理员.....	3734
12.13.2 创建或更新安全管理员.....	3736



12.14 动态数据脱敏接口.....	3738
12.14.1 查询动态数据脱敏策略列表.....	3738
12.14.2 创建数据脱敏策略.....	3744
12.14.3 查询某个脱敏策略的详细信息.....	3762
12.14.4 更新动态数据脱敏策略.....	3769
12.14.5 批量删除动态脱敏策略.....	3787
12.15 权限审批接口.....	3791
12.15.1 获取表权限审批人列表.....	3791
12.15.2 提交表权限申请.....	3794
12.15.3 审批通过工单.....	3797
12.15.4 驳回工单.....	3798
12.15.5 获取审批工单列表.....	3800
12.16 权限申请接口.....	3804
12.16.1 撤回工单申请.....	3804
<b>13 应用示例.....</b>	<b>3807</b>
13.1 数据集成应用示例.....	3807
13.2 数据开发应用示例.....	3815
<b>14 附录.....</b>	<b>3816</b>
14.1 公共消息头.....	3816
14.2 解析响应消息体中的流.....	3818
14.3 状态码.....	3818
14.4 错误码.....	3821
14.4.1 数据集成 API 错误码.....	3821
14.4.2 数据开发错误码.....	3838

# 1 使用前必读

## 1.1 概述

欢迎使用数据治理中心DataArts Studio服务。DataArts Studio是针对企业数字化运营诉求提供的数据全生命周期管理、具有智能数据管理能力的一站式治理运营平台，包含数据集成、数据架构、数据开发、数据质量、数据目录、数据服务等功能，支持行业知识库智能化建设，支持大数据存储、大数据计算分析引擎等数据底座，帮助企业快速构建从数据接入到数据分析的端到端智能数据系统，消除数据孤岛，统一数据标准，加快数据变现，实现数字化转型。

您可以使用本文档提供的API对数据治理中心的各组件进行相关操作，如创建集群、创建连接等。支持的全部操作请参见[API概览](#)。

在调用数据治理中心API之前，请确保已经充分了解数据治理中心相关概念，详细信息请参见[产品介绍](#)。

## 1.2 调用说明

数据治理中心提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用。

调用方法请参见[如何调用API](#)。

- [构造请求](#)
- [认证鉴权](#)
- [返回结果](#)

## 1.3 基本概念

- 账号  
用户的账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用用户进行日常管理工作。
- 用户

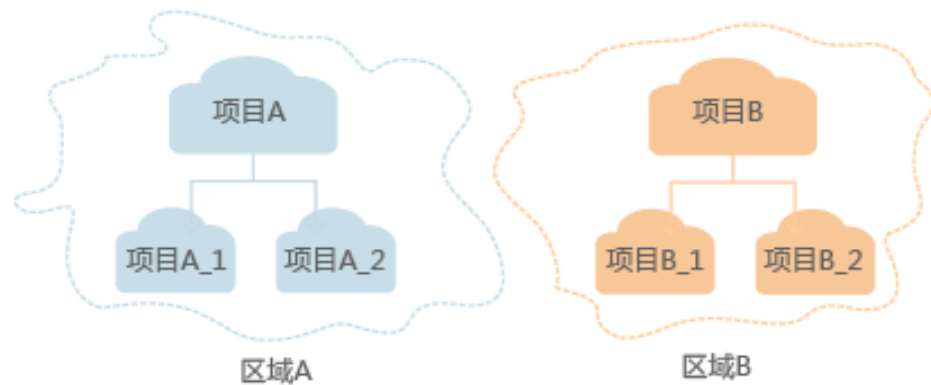
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

在**项目ID和账号ID**下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 项目

区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



## 1.4 终端节点 Endpoint

### 获取终端节点

终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同。

数据治理中心的终端节点Endpoint构造规则如下，请您根据业务需要选择对应区域的终端节点。

- CDM数据集成接口：cdm.**{region\_id}**.myhuaweicloud.com，具体如**表1-1**所示。
- DLF数据开发V1接口：dayu-dlf.**{region\_id}**.myhuaweicloud.com，具体如**表1-2**所示。
- DLF数据开发V2接口：dayu.**{region\_id}**.myhuaweicloud.com，具体如**表1-2**所示。
- DataArts Studio其他组件接口（如管理中心、数据架构、数据质量、数据目录、数据服务、数据安全、数据地图等）：dayu.**{region\_id}**.myhuaweicloud.com，具体如**表1-3**所示。

表 1-1 CDM 数据集成 Endpoint

区域名称	区域ID	终端节点 ( Endpoint )	协议类型
华北-北京一	cn-north-1	cdm.cn-north-1.myhuaweicloud.com	HTTPS
华北-北京二	cn-north-2	cdm.cn-north-2.myhuaweicloud.com	HTTPS
华北-北京四	cn-north-4	cdm.cn-north-4.myhuaweicloud.com	HTTPS
华北-乌兰察布一	cn-north-9	cdm.cn-north-9.myhuaweicloud.com	HTTPS
华东-上海一	cn-east-3	cdm.cn-east-3.myhuaweicloud.com	HTTPS
华东-上海二	cn-east-2	cdm.cn-east-2.myhuaweicloud.com	HTTPS
华南-广州	cn-south-1	cdm.cn-south-1.myhuaweicloud.com	HTTPS
西南-贵阳一	cn-southwest-2	cdm.cn-southwest-2.myhuaweicloud.com	HTTPS

表 1-2 数据开发 Endpoint

区域名称	区域ID	V1接口终端节点 ( Endpoint )	V2接口终端节点 ( Endpoint )	协议类型
华北-北京一	cn-north-1	dayu-dlf.cn-north-1.myhuaweicloud.com	dayu.cn-north-1.myhuaweicloud.com	HTTPS
华北-北京二	cn-north-2	dayu-dlf.cn-north-2.myhuaweicloud.com	dayu.cn-north-2.myhuaweicloud.com	HTTPS
华北-北京四	cn-north-4	dayu-dlf.cn-north-4.myhuaweicloud.com	dayu.cn-north-4.myhuaweicloud.com	HTTPS
华北-乌兰察布一	cn-north-9	dayu-dlf.cn-north-9.myhuaweicloud.com	dayu.cn-north-9.myhuaweicloud.com	HTTPS
华东-上海一	cn-east-3	dayu-dlf.cn-east-3.myhuaweicloud.com	dayu.cn-east-3.myhuaweicloud.com	HTTPS
华东-上海二	cn-east-2	dayu-dlf.cn-east-2.myhuaweicloud.com	dayu.cn-east-2.myhuaweicloud.com	HTTPS
华南-广州	cn-south-1	dayu-dlf.cn-south-1.myhuaweicloud.com	dayu.cn-south-1.myhuaweicloud.com	HTTPS

区域名称	区域ID	V1接口终端节点 (Endpoint)	V2接口终端节点 (Endpoint)	协议类型
西南-贵阳一	cn-southwest-2	dayu-dlf.cn-southwest-2.myhuaweicloud.com	dayu.cn-southwest-2.myhuaweicloud.com	HTTPS

表 1-3 其他组件 Endpoint

区域名称	区域ID	终端节点 (Endpoint)	协议类型
华北-北京一	cn-north-1	dayu.cn-north-1.myhuaweicloud.com	HTTPS
华北-北京二	cn-north-2	dayu.cn-north-2.myhuaweicloud.com	HTTPS
华北-北京四	cn-north-4	dayu.cn-north-4.myhuaweicloud.com	HTTPS
华北-乌兰察布一	cn-north-9	dayu.cn-north-9.myhuaweicloud.com	HTTPS
华东-上海一	cn-east-3	dayu.cn-east-3.myhuaweicloud.com	HTTPS
华东-上海二	cn-east-2	dayu.cn-east-2.myhuaweicloud.com	HTTPS
华南-广州	cn-south-1	dayu.cn-south-1.myhuaweicloud.com	HTTPS
西南-贵阳一	cn-southwest-2	dayu.cn-southwest-2.myhuaweicloud.com	HTTPS

## 1.5 项目 ID 和账号 ID

### 获取项目 ID 和账号 ID

项目ID表示租户的资源，账号ID对应当前账号，IAM用户ID对应当前用户。用户可在对应页面下查看不同Region对应的项目ID、账号ID和用户ID。

1. 注册并登录管理控制台。
2. 在用户名的下拉列表中单击“我的凭证”。
3. 在“API凭证”页面，查看账号名和账号ID、IAM用户名和IAM用户ID，在项目列表中查看项目和项目ID。

### 调用 API 获取项目 ID

项目ID可以通过调用[查询指定条件下的项目信息](#)API获取，接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可参考IAM文档获取。

接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，其中projects下的“id”即为项目ID。当返回多个id，请依据实际的区域（name）获取。

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "region-name",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

## 1.6 实例 ID 和工作空间 ID

### 获取 DataArts Studio 实例 ID 和工作空间 ID

DataArts Studio的实例ID和工作空间ID可以从DataArts Studio控制台的URI链接中获取。

1. 在DataArts Studio控制台首页，选择对应工作空间，并单击任一模块，如“管理中心”。
2. 进入管理中心页面后，从浏览器地址栏中获取“instanceId”和“workspace”对应的值，即为DataArts Studio的实例ID和工作空间ID。

如图1-2所示，实例ID为**6b88...2688**，工作空间ID为**1dd3bc...d93f0**。

图 1-2 获取实例 ID 和工作空间 ID



dayu/workspace=1dd3bc...1d93f0&instanceId=6b88...2688

## 1.7 数据资产 guid

每个业务资产、技术资产或指标资产都具备guid，guid是资产的唯一标识符。在调用数据目录或数据地图接口时，部分URI中需要填入guid。

资产guid可通过相应查询接口获取，也可以通过控制台获取，推荐您通过相应查询接口获取。

控制台获取操作相对复杂，如下所示，为您介绍如何通过控制台获取guid资产：

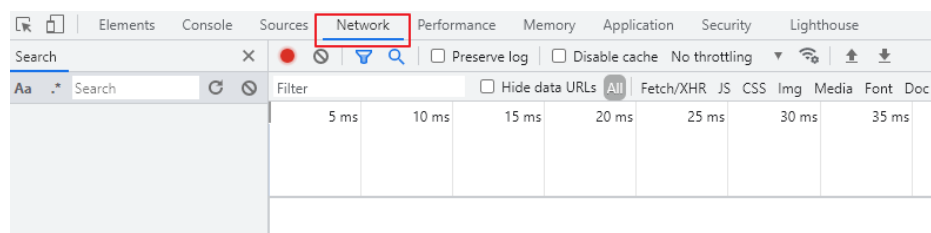
1. 在DataArts Studio控制台首页，选择实例，单击“进入控制台”，选择“数据地图”或对应工作空间的“数据目录”，进入数据地图或数据目录页面。

图 1-3 选择数据地图或数据目录



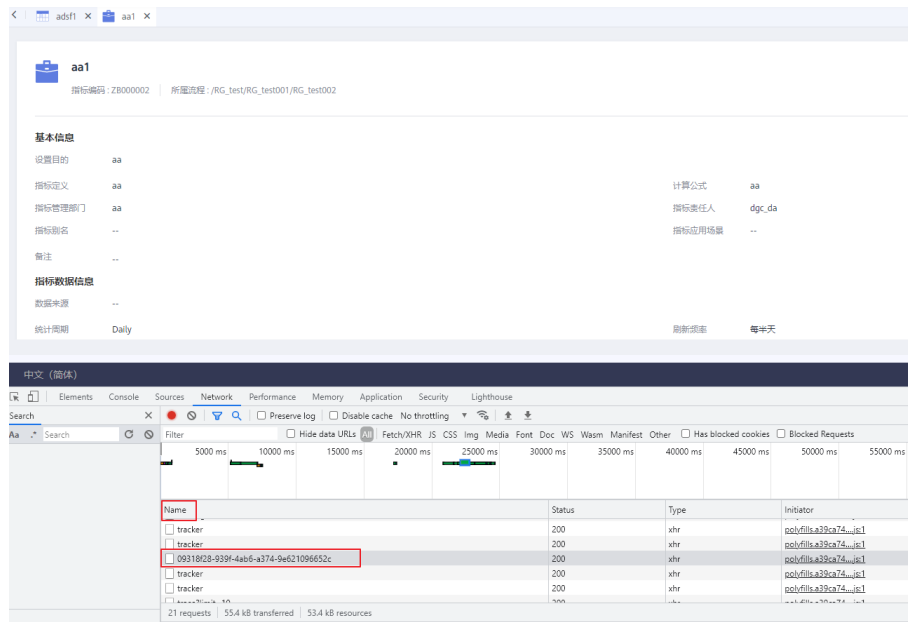
2. 按下F12，打开开发者调试工具，然后选择Network功能。

图 1-4 选择 Network



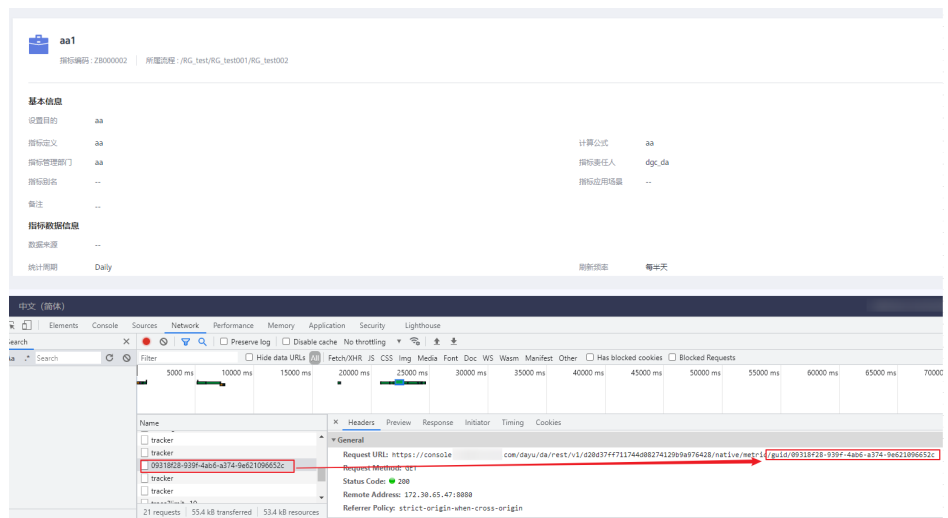
3. 在数据地图的“首页”界面或“数据搜索”界面，或者在数据目录的“总览”界面或“数据目录”界面，选择需要查询guid的资产，并单击资产名进入资产详情页。
  - a. 在Network请求中，寻找Name形如“09318f28-939f-4ab6-a374-9e621096652c”的长字符串。

图 1-5 寻找长字符串



- b. 单击该字符串，弹出具体请求的弹窗。在Request URL中，可以确认该字符串即为该资产的guid。

图 1-6 获取 guid



## 1.8 约束与限制

- 仅当创建IAM用户时的访问方式勾选“编程访问”后，此IAM用户才能通过认证鉴权，从而使用API、SDK等方式访问DataArts Studio。
- 各组件API的流量限制请参见[API概览](#)。
- 数据集成API的约束如下：
  - CDM的作业数据量太多，会造成数据库压力，建议定时清理不需要的作业。
  - 短时间内下发大量作业，可能会导致集群资源耗尽异常等，您调用API时需要注意。



- CDM是批量离线迁移工具，不建议客户创建大量小作业场景。
- 其他组件API更详细的限制请参见具体API的说明。

# 2 API 概览

## 2.1 数据集成 API 概览

表 2-1 数据集成 API 类型

类型	说明
集群管理	集群管理类型接口。
作业管理	作业管理类型接口。
连接管理	连接管理类型接口。

### 集群管理

表 2-2 集群管理

API	说明	流控策略
查询集群详情	查询集群详情接口。	cdm_common
删除集群	删除集群接口。	cdm_b
查询所有可用区	查询CDM集群的所有可用区。	cdm_common
查询支持的版本	查询CDM集群支持的版本。	cdm_c
查询版本规格	按版本ID查询所有兼容规格。	cdm_common
查询规格详情	查询指定规格ID的规格详情。	cdm_common
查询所有集群的企业项目ID	查询当前项目下的所有集群的企业项目ID。	cdm_common

API	说明	流控策略
<a href="#">查询集群的企业项目ID</a>	查询指定集群的企业项目ID。	cdm_common
<a href="#">查询集群实例信息</a>	查询集群实例信息。	cdm_common
<a href="#">修改集群</a>	修改CDM集群配置。	cdm_common
<a href="#">重启集群</a>	重启集群接口。	cdm_b
<a href="#">启动集群</a>	启动集群接口。	cdm_b
<a href="#">停止集群（待下线）</a>	停止集群接口。	cdm_b
<a href="#">创建集群</a>	创建集群接口。	cdm_a
<a href="#">查询集群列表</a>	查询集群列表接口。	cdm_common

## 作业管理

表 2-3 作业管理

API	说明	流控策略
<a href="#">查询作业</a>	查询作业接口。	cdm_common
<a href="#">删除作业</a>	删除作业接口。	cdm_job
<a href="#">修改作业</a>	修改作业接口。	cdm_job_create
<a href="#">随机集群创建作业并执行</a>	随机集群创建作业并执行接口。	cdm_common
<a href="#">停止作业</a>	停止作业接口。	cdm_job
<a href="#">指定集群创建作业</a>	指定集群创建作业接口。	cdm_job_create
<a href="#">启动作业</a>	启动作业接口。	cdm_job_start
<a href="#">查询作业状态</a>	查询作业状态接口。	cdm_d
<a href="#">查询作业执行历史</a>	查询作业执行历史接口。	cdm_common

## 连接管理

表 2-4 连接管理

API	说明	流控策略
<a href="#">创建连接</a>	创建连接接口。	cdm_common
<a href="#">查询连接</a>	查询连接接口。	cdm_common

API	说明	流控策略
<a href="#">删除连接</a>	删除连接接口。	<a href="#">cdm_common</a>
<a href="#">修改连接</a>	修改连接接口。	<a href="#">cdm_common</a>

## 参考：流控策略详情

表 2-5 数据集成 API 流控策略

流控策略名称	策略类型	API流量限制 (次/min)	用户流量限制 (次/min)
cdm_a	共享流控 (此策略所有接口共享流量限制)	100	5
cdm_b	共享流控 (此策略所有接口共享流量限制)	200	20
cdm_c	共享流控 (此策略所有接口共享流量限制)	6000	50
cdm_d	共享流控 (此策略所有接口共享流量限制)	6000	120
cdm_job	共享流控 (此策略所有接口共享流量限制)	6000	1200
cdm_job_create	共享流控 (此策略所有接口共享流量限制)	6000	1200
cdm_job_start	共享流控 (此策略所有接口共享流量限制)	6000	1200
cdm_common	共享流控 (此策略所有接口共享流量限制)	6000	120

## 2.2 数据开发 API (V1) 概览

表 2-6 数据开发 API (V1) 类型

类型	说明
连接管理 (待下线)	连接管理类型接口。 <b>说明</b> 连接管理能力由管理中心统一提供, 相关接口已不再维护, 推荐使用管理中心API进行连接管理。
脚本开发	脚本开发类型接口。
资源管理	资源管理类型接口。
作业开发	作业开发类型接口。

### 连接管理 (待下线)

表 2-7 连接管理 (待下线)

API	说明	API流量限制 (次/s)	用户流量限制 (次/s)
创建连接 (待下线)	创建一个新连接。	-	-
查询连接列表 (待下线)	查询连接列表。	-	-
查询连接详情 (待下线)	查询指定连接的配置详情。	-	-
修改连接 (待下线)	编辑修改连接的配置。	-	-
删除连接 (待下线)	删除一个连接。	-	-
导出连接 (待下线)	导出zip压缩包格式的连接文件, 包括所有连接的定义。	-	-
导入连接 (待下线)	将存放在OBS桶的连接文件导入到数据开发模块中, 一次可以导入一个或多个连接。	-	-

## 脚本开发

表 2-8 脚本开发

API	说明	API流量限制 (次/s)	用户流量限制 (次/s)
<a href="#">创建脚本</a>	创建一个新脚本。	300	30
<a href="#">修改脚本内容</a>	本接口用以修改脚本的配置项或者脚本内容。	300	30
<a href="#">查询脚本信息</a>	此接口可以查询指定脚本的定义，包括脚本类型和脚本内容。	300	30
<a href="#">查询脚本列表</a>	此接口用以查询脚本列表。	300	30
<a href="#">查询脚本实例执行结果</a>	该接口用以获取脚本实例的执行状态和结果。	300	30
<a href="#">删除脚本</a>	此接口用以删除指定脚本。	300	30
<a href="#">执行脚本</a>	该接口用以执行指定脚本，脚本每执行一次，生成一个脚本实例。	300	30
<a href="#">停止脚本实例的执行</a>	停止脚本实例的执行。	300	30

## 资源管理

表 2-9 资源管理

API	说明	API流量限制 (次/s)	用户流量限制 (次/s)
<a href="#">创建资源</a>	创建资源。	100	10
<a href="#">查询资源列表</a>	查询资源列表。	100	10
<a href="#">修改资源</a>	修改资源。	100	10
<a href="#">删除资源</a>	删除资源。	100	10
<a href="#">查询资源详情</a>	查询资源详情。	300	30

## 作业开发

表 2-10 作业开发

API	说明	API流量限制 (次/s)	用户流量限制 (次/s)
<a href="#">创建作业</a>	创建作业。	300	30
<a href="#">修改作业</a>	更新作业。	300	30
<a href="#">查询作业列表</a>	查询作业列表清单。	300	30
<a href="#">查询作业详情</a>	查询作业详情。	100	10
<a href="#">查询作业文件</a>	查询作业文件。	100	10
<a href="#">导出作业</a>	导出作业。	100	10
<a href="#">批量导出作业</a>	批量导出作业。	100	10
<a href="#">导入作业</a>	导入作业。	100	10
<a href="#">立即执行作业</a>	立即执行作业。	300	30
<a href="#">启动作业</a>	启动作业。	300	30
<a href="#">停止作业</a>	停止作业。	300	30
<a href="#">删除作业</a>	删除作业。	300	30
<a href="#">停止作业实例</a>	停止作业实例。	300	30
<a href="#">重跑作业实例</a>	重跑作业实例。	300	30
<a href="#">查询实时作业运行状态</a>	查询实时作业的运行状态。	300	30
<a href="#">查询作业实例列表</a>	查询作业实例列表。	300	30
<a href="#">查询作业实例详情</a>	查询作业实例详情。	300	30
<a href="#">查询系统任务详情</a>	查询异步任务详情。	100	10

## 2.3 数据开发 API (V2) 概览

表 2-11 数据开发 API (V2) 类型

类型	说明
<a href="#">作业开发</a>	作业开发类型接口。

## 作业开发

表 2-12 作业开发

API	说明	API流量限制 (次/s)	用户流量限制 (次/s)
<a href="#">创建补数据实例</a>	创建一个补数据实例。	100	10
<a href="#">查询补数据实例</a>	查询补数据实例列表，支持分页查询。	100	10
<a href="#">停止补数据实例</a>	停止指定的补数据实例。	100	10
<a href="#">修改作业名称</a>	对作业名称进行重命名操作。	100	10
<a href="#">查询发布包列表</a>	用于查询发布包的列表信息。	100	10
<a href="#">查询指定发布包详情</a>	用于查询指定的发布包详情信息。	100	10
<a href="#">设置作业标签</a>	此设置作业标签。	100	10
<a href="#">查询告警通知记录</a>	查询指定时间段内的告警通知记录。	100	10
<a href="#">发布任务包</a>	目前支持发布包管理中相关包的发布，支持发布多个任务包。	100	10
<a href="#">撤销任务包</a>	目前支持发布包管理中相关包的撤销，支持撤销多个任务包。	100	10
<a href="#">查询实例运行状态</a>	查询实例运行状态，包括当天、昨天、前天以及近7天的实例状态情况。	1	1
<a href="#">查询任务完成情况</a>	查询任务完成情况。包括今天，昨天以及近7天的平均任务完成情况。	1	1
<a href="#">查询指定作业的实例列表</a>	根据作业名称精确查询作业实例列表。	100	10
<a href="#">重跑作业实例</a>	支持重跑作业实例以及上下游的作业实例。	100	10



## 2.4 管理中心 API 概览

表 2-13 管理中心 API 类型

类型	说明
<a href="#">数据连接管理</a>	数据连接管理类型接口。
<a href="#">购买实例</a>	购买实例类型接口。
<a href="#">工作空间管理</a>	工作空间管理类型接口。
<a href="#">实例管理</a>	实例管理类型接口。
<a href="#">工作空间用户管理</a>	工作空间用户管理类型接口。
<a href="#">数据源元数据获取</a>	数据源元数据获取类型接口。
<a href="#">实例规格变更</a>	实例规格变更类型接口。

### 数据连接管理

表 2-14 数据连接管理

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查询数据连接列表</a>	查询数据连接列表。	12000	60
<a href="#">创建数据连接</a>	创建数据连接。	6000	40
<a href="#">测试创建数据连接</a>	测试创建数据连接。	3000	20
<a href="#">查询单个数据连接信息</a>	查询单个数据连接信息。	12000	60
<a href="#">更新数据连接信息</a>	更新数据连接信息。	6000	40
<a href="#">删除数据连接</a>	删除数据连接。	6000	40

### 购买实例

表 2-15 购买实例

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">DataArtsStudio实例一键购买接口</a>	DataArtsStudio实例一键购买接口。	12000	60

## 工作空间管理

表 2-16 工作空间管理

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取工作空间列表</a>	获取工作空间列表。	3000	20
<a href="#">创建工作空间</a>	创建工作空间。	3000	20
<a href="#">获取单个工作空间信息</a>	获取单个工作空间信息。	3000	20

## 实例管理

表 2-17 实例管理

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取实例列表</a>	获取实例列表。	3000	20

## 工作空间用户管理

表 2-18 工作空间用户管理

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取工作空间用户角色</a>	获取工作空间用户角色。	3000	20
<a href="#">编辑工作空间用户或用户组</a>	编辑工作空间用户或用户组。	3000	20
<a href="#">获取工作空间用户信息</a>	获取工作空间用户信息。	3000	20
<a href="#">添加工作空间用户</a>	添加工作空间用户。	3000	20
<a href="#">删除工作空间用户</a>	删除工作空间用户。	3000	20

## 数据源元数据获取

表 2-19 数据源元数据获取

API	说明	API流量限制（次/min）	用户流量限制（次/min）
<a href="#">获取数据库列表</a>	获取数据库列表。	12000	60
<a href="#">获取schemas</a>	获取schemas，目前只有DWS和采用postgresql驱动的RDS数据源支持schema，请在调用前确认该数据源是否支持schema字段。	12000	60
<a href="#">获取数据源中的表</a>	获取数据源中的表。	12000	60
<a href="#">获取数据源中表的字段</a>	获取数据源中表的字段。	12000	60

## 实例规格变更

表 2-20 实例规格变更

API	说明	API流量限制（次/min）	用户流量限制（次/min）
<a href="#">规格变更接口</a>	规格变更接口。	3000	20

## 2.5 数据架构 API 概览

表 2-21 数据架构 API 类型

类型	说明
<a href="#">概览</a>	概览接口。
<a href="#">信息架构接口</a>	信息架构接口。
<a href="#">数据标准接口</a>	数据标准接口
<a href="#">数据源接口</a>	数据源接口。
<a href="#">码表管理接口</a>	码表管理接口。
<a href="#">流程架构接口</a>	流程架构接口。
<a href="#">数据标准模板接口</a>	数据标准模板接口。

类型	说明
审批管理接口	审批管理接口。
主题管理接口	主题管理接口。
主题层级接口	主题层级接口。
目录管理	目录管理，码表和数据标准的目录。
原子指标接口	原子指标接口。
衍生指标接口	衍生指标接口。
复合指标接口	复合指标接口。
维度接口	维度接口。
限定接口	限定接口，包含时间限定、通用限定。
维度表接口	维度表接口。
事实表接口	事实表接口。
汇总表接口	汇总表接口。
业务指标接口	业务指标接口。
版本信息接口	版本信息接口。
关系建模接口	表模型。
导入导出接口	导入导出接口。
自定义项接口	自定义项接口。
标签接口	标签接口。
质量规则接口	质量规则接口。

## 概览

表 2-22 概览

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
总览统计信息	总览统计信息。	21600	240
模型统计信息	单个模型中的统计信息。	21600	240
关系建模统计信息	关系建模页面，外层的统计信息。	21600	240

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">标准覆盖率统计信息</a>	查看某个数据标准在所有模型字段中的覆盖率，即使用该标准的字段占总字段的百分比。	21600	240

## 信息架构接口

表 2-23 信息架构接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查询多种类型的表信息</a>	从信息架构中查询多种类型的表信息，包括逻辑实体、物理表、维度表、事实表、汇总表。	21600	240

## 数据标准接口

表 2-24 数据标准接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取数据标准集合</a>	根据查询条件分页获取数据标准集合，按修改时间降序排序。	21600	240
<a href="#">创建数据标准</a>	创建数据标准。	10800	120
<a href="#">删除数据标准</a>	删除数据标准。	10800	120
<a href="#">修改数据标准</a>	修改数据标准。	10800	120
<a href="#">查看数据标准详情</a>	通过ID获取数据标准详情信息。	21600	240
<a href="#">关联属性与数据标准</a>	关联属性与数据标准。	10800	120

## 数据源接口

表 2-25 数据源接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取数据连接信息</a>	获取指定类型下的数据连接信息。	21600	240

## 码表管理接口

表 2-26 码表管理接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查询码表列表</a>	查询码表列表。	21600	240
<a href="#">创建码表</a>	创建码表。	10800	120
<a href="#">删除码表</a>	删除码表。	10800	120
<a href="#">修改码表</a>	修改码表。	10800	120
<a href="#">查看码表详情</a>	通过ID查看码表的详情信息。	21600	240
<a href="#">查看码表字段值</a>	查看码表字段值。	21600	240
<a href="#">编辑码表字段值</a>	编辑码表字段值。	10800	120

## 流程架构接口

表 2-27 流程架构接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取所有流程架构目录树</a>	获取所有目录树。	21600	240
<a href="#">查询流程架构列表</a>	查询流程架构列表。	21600	240
<a href="#">创建流程架构</a>	创建流程架构。	10800	120
<a href="#">修改流程架构</a>	修改流程架构。	10800	120
<a href="#">删除流程架构</a>	删除流程架构。	10800	120

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找流程架构详情</a>	查找流程架构详情。	21600	240

## 数据标准模板接口

表 2-28 数据标准模板接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查询数据标准模板</a>	查询当前工作空间下的数据标准模板。	21600	240
<a href="#">创建数据标准模板</a>	创建当前工作空间下的数据标准模板自定义项。	10800	120
<a href="#">修改数据标准模板</a>	修改数据标准模板。	10800	120
<a href="#">删除数据标准模板</a>	删除数据标准模板。	10800	120
<a href="#">初始化数据标准模板</a>	初始化数据标准模板。	10800	120

## 审批管理接口

表 2-29 审批管理接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取审批单</a>	获取审批单。	21600	240
<a href="#">撤回审批单</a>	撤回审批单。	10800	120
<a href="#">审批单处理</a>	审批驳回/通过，单个或多个action-id=reject/resolve。	10800	120
<a href="#">批量发布</a>	批量发布。	10800	120
<a href="#">批量下线</a>	批量下线。	10800	120
<a href="#">创建审批人</a>	创建审批人。	10800	120
<a href="#">查询审批人列表</a>	查询审批人列表。	21600	240
<a href="#">删除审批人</a>	删除审批人。	10800	120

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">删除实体下展</a>	当已发布的实体被编辑时，其会生成下展，该接口用于删除实体的下展信息。	10800	120
<a href="#">获取下展信息与已发布实体的差异</a>	当已发布的实体被编辑时，其会生成下展，该接口用于获取下展信息与已发布实体的差异。	21600	240

## 主题管理接口

表 2-30 主题管理接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找主题列表</a>	通过名称（支持模糊查询）、创建者、责任人、状态、修改时间分页查找主题。	21600	240
<a href="#">删除主题</a>	删除主题。	10800	120
<a href="#">创建主题</a>	创建主题。	10800	120
<a href="#">修改主题</a>	修改主题。	10800	120
<a href="#">获取主题树信息</a>	获取数据资产主题树信息 l1, l2, l3。	21600	240
<a href="#">查找主题列表(新)</a>	查找主题(新)。	21600	240
<a href="#">删除主题(新)</a>	删除主题(新)。	10800	120
<a href="#">创建主题(新)</a>	创建主题(新)。	10800	120
<a href="#">修改主题(新)</a>	修改主题(新)。	10800	120

## 主题层级接口

表 2-31 主题层级接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取主题层级</a>	获取主题层级。	21600	240



API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">修改或删除主题层级</a>	修改或删除主题层级。	10800	120

## 目录管理

表 2-32 目录管理

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取所有目录</a>	获取所有目录(数据标准、码表)。	21600	240
<a href="#">创建目录</a>	创建目录(数据标准、码表)。	10800	120
<a href="#">修改目录</a>	修改目录(数据标准、码表)。	10800	120
<a href="#">删除目录</a>	删除目录(数据标准、码表)。	10800	120

## 原子指标接口

表 2-33 原子指标接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找原子指标</a>	通过中英文名称、创建者、审核人、状态、修改时间分页查找原子指标信息看，中英文名称支持模糊查询。	21600	240
<a href="#">新建原子指标</a>	新建单个原子指标。	10800	120
<a href="#">更新原子指标</a>	更新单个原子指标。	10800	120
<a href="#">删除原子指标</a>	批量删除原子指标。	10800	120
<a href="#">查看原子指标详情</a>	通过ID获取原子指标详情信息。	21600	240

## 衍生指标接口

表 2-34 衍生指标接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找衍生指标</a>	通过中英文名称、创建者、审核人、状态、修改时间、l3Id分页查找衍生指标信息。	21600	240
<a href="#">新建衍生指标</a>	根据参数，新建衍生指标指标。	10800	120
<a href="#">更新衍生指标</a>	根据传入参数，更新衍生指标。	10800	120
<a href="#">删除衍生指标</a>	根据衍生指标ID，删除衍生指标。	10800	120
<a href="#">查看衍生指标详情</a>	通过ID获取衍生详情信息。	21600	240

## 复合指标接口

表 2-35 复合指标接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找复合指标</a>	通过中英文名称、创建者、审核人、状态、修改时间、l3Id分页查找复合指标信息。	21600	240
<a href="#">新建复合指标</a>	根据参数，新建复合指标。	10800	120
<a href="#">更新复合指标</a>	根据参数，更新复合指标。	10800	120
<a href="#">删除复合指标</a>	根据ID集合删除复合指标。	10800	120
<a href="#">查看复合指标详情</a>	通过ID获取复合指标详情信息。	21600	240

## 维度接口

表 2-36 维度接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找维度</a>	通过中英文名称、描述、创建者、审核人、状态、l3Id、衍生指标idList、修改时间分页查找维度信息。	21600	240
<a href="#">新建维度</a>	根据参数新建维度。	10800	120
<a href="#">更新维度</a>	根据参数，更新维度信息。	10800	120
<a href="#">删除维度</a>	根据传入的维度ID，删除维度。	10800	120
<a href="#">查看维度详情</a>	通过ID查看维度详情信息。	21600	240
<a href="#">查看维度颗粒度</a>	查询维度颗粒度，依据tableId查询涉及所有维度，不传tableId查询所有维度组颗粒度。	21600	240
<a href="#">查看逆向维度表任务</a>	查看逆向维度表任务。	21600	240

## 限定接口

表 2-37 限定接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找业务限定</a>	通过中英文名称、描述、创建者、审核人、限定分组id、修改时间状态分页查找限定信息。	21600	240
<a href="#">查看限定详情</a>	通过ID查看限定详情信息。	21600	240

## 维度表接口

表 2-38 维度表接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找维度表</a>	通过中英文名称、创建者、审核人、状态、修改时间分页查找维度表信息。	21600	240
<a href="#">删除维度表</a>	根据维度表ID，删除维度表。	10800	120
<a href="#">查看维度表详情</a>	通过ID查看维度表的详情信息。	21600	240

## 事实表接口

表 2-39 事实表接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找事实表</a>	通过中英文名称、创建者、审核人、状态、修改时间分页查找事实表信息。	21600	240
<a href="#">删除事实表</a>	根据ID集合删除事实表。	10800	120
<a href="#">查看事实表详情</a>	通过ID查看事实表的详情信息。	21600	240
<a href="#">查看逆向事实表任务</a>	查看逆向事实表任务。	21600	240

## 汇总表接口

表 2-40 汇总表接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找汇总表</a>	通过中英文名称、创建者、审核人、状态、修改时间分页查找汇总表信息，中英文名称支持模糊查询。	21600	240

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">新建汇总表</a>	根据入参，手动创建汇总表。	10800	120
<a href="#">更新汇总表</a>	更新汇总表。	10800	120
<a href="#">删除汇总表</a>	批量删除汇总表，只能删除状态为草稿、已线下、已驳回的表。	10800	120
<a href="#">查看汇总表详情</a>	通过ID查看汇总表的详情信息。	21600	240

## 业务指标接口

表 2-41 业务指标接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查询业务指标信息</a>	通过名称、创建者、修改时间分页查找业务指标信息。	21600	240
<a href="#">创建业务指标</a>	创建业务指标。	10800	120
<a href="#">更新业务指标</a>	更新业务指标。	10800	120
<a href="#">删除业务指标</a>	删除业务指标。	10800	120
<a href="#">查看指标详情</a>	通过ID查看指标的详情信息。	21600	240
<a href="#">查看指标维度信息</a>	查看指标维度信息。	21600	240
<a href="#">查看指标责任人信息</a>	查看指标责任人信息。	21600	240
<a href="#">获取指标关联信息</a>	获取当前指标图谱。	21600	240

## 版本信息接口

表 2-42 版本信息接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找版本信息</a>	通过名称、创建者、修改时间查找版本信息。	21600	240
比较版本信息	通过两个版本id, 比较两者差异。	60	24

## 关系建模接口

表 2-43 关系建模接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查找表模型列表</a>	通过中英文名称、创建者、审核人、状态、修改时间分页查找关系建模中的表模型信息, 包括逻辑实体、物理表和其属性。	21600	240
<a href="#">创建表模型</a>	在关系建模中创建一个表模型, 包括逻辑实体和物理表。	10800	120
<a href="#">更新表模型</a>	在关系建模中更新一个表模型及其属性, 包括逻辑实体和物理表。	10800	120
<a href="#">删除表模型</a>	在关系建模中删除一个表模型及其属性, 包括逻辑实体和物理表。	10800	120
<a href="#">查询关系</a>	通过关系名称(支持模糊查询)、创建人、开始时间、结束时间等分页查找关系信息。	21600	240
<a href="#">查看关系详情</a>	通过ID获取关系详情信息。	21600	240
<a href="#">查询模型下所有关系</a>	查询模型下所有关系。	21600	240
<a href="#">查看表模型详情</a>	通过ID获取模型表详情信息。	21600	240

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">获取模型</a>	获取当前空间下的全部模型信息。	21600	240
<a href="#">新建模型工作区</a>	新建模型工作区。	10800	120
<a href="#">更新模型工作区</a>	更新模型工作区。	10800	120
<a href="#">删除模型工作区</a>	删除模型工作区。	10800	120
<a href="#">查询模型详情</a>	查询物理模型或逻辑模型的工作区空间详情。	21600	240
<a href="#">查询目的表和字段(待下线)</a>	查询目的表和字段(待下线)。	21600	240
<a href="#">导出模型中表的DDL语句</a>	根据模型ID导出指定表的DDL语句。	10	4
<a href="#">转换逻辑模型为物理模型</a>	转换逻辑模型为物理模型，转换成功则显示转换后的目标模型信息。	10800	120
<a href="#">获取操作结果</a>	获取批量操作的结果，如逻辑模型转物理模型和逆向数据库操作。	21600	240

## 导入导出接口

表 2-44 导入导出接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">导入主题</a>	用于导入主题。	10	4
<a href="#">导入模型，关系建模，维度建模，码表，业务指标以及流程架构</a>	导入模型，关系建模，维度建模，码表，业务指标以及流程架构。	10	4
<a href="#">查询导入结果</a>	查询导入excel的处理结果（其中参数uuid获取为：/design/models/action或/design/catalogs/action接口返回结果）。	21600	240

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">导出业务数据</a>	根据请求参数，导出业务数据，可以导出：码表、数据标准、原子指标、衍生指标、复合指标、汇总表、业务指标、主题、流程、逻辑模型、物理模型、维度、事实表。	10	4
<a href="#">获取excel导出结果</a>	根据请求导出业务数据（/export-model）时返回的uuid，获取excel导出结果。	21600	240

## 自定义项接口

表 2-45 自定义项接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">查询自定义项</a>	查询自定义项（包括表自定义项、属性自定义项、主题自定义项、业务指标自定义项）。	21600	240
<a href="#">修改自定义项</a>	修改自定义项（包括表自定义项、属性自定义项、主题自定义项、业务指标自定义项）。	10800	120

## 标签接口

表 2-46 标签接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">添加标签</a>	根据资产（表或属性）的ID给资产打上标签。	10800	120
<a href="#">删除标签</a>	根据资产（表或属性）的ID删除资产标签。	10800	120



## 质量规则接口

表 2-47 质量规则接口

API	说明	API流量限制 (次/min)	用户流量限制 (次/min)
<a href="#">更新表的异常数据输出配置</a>	更新表的异常数据输出配置，包括是否生成异常数据、设置异常数据数据库或 Schema、设置异常表表前缀/表后缀。	10800	120
<a href="#">清空质量规则</a>	清空表的质量规则。	10800	120

## 2.6 数据质量 API 概览

表 2-48 数据质量 API 类型

类型	说明
<a href="#">目录接口</a>	目录接口。
<a href="#">规则模板接口</a>	规则模板接口。
<a href="#">质量作业接口</a>	质量作业接口。
<a href="#">对账作业接口</a>	对账作业接口。
<a href="#">运维管理接口</a>	运维管理接口。
<a href="#">任务实例接口</a>	任务实例接口。
<a href="#">质量报告接口</a>	质量报告接口。
<a href="#">导入导出接口</a>	导入导出接口。

### 目录接口

表 2-49 目录接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取作业目录</a>	获取作业目录。	300	40

## 规则模板接口

表 2-50 规则模板接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取规则模板列表</a>	分页获取规则模板列表。	300	40
<a href="#">创建规则模板</a>	创建规则模板。	300	40
<a href="#">获取规则模板详情</a>	获取规则模板详情。	300	40
<a href="#">更新规则模板</a>	更新规则模板	300	40
<a href="#">批量删除规则模板</a>	批量删除规则模板。	300	40

## 质量作业接口

表 2-51 质量作业接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取质量作业列表 V1</a>	批量查询数据质量作业，返回质量作业列表。	300	40
<a href="#">获取质量作业列表</a>	获取质量作业列表。	300	40
<a href="#">获取质量作业详情</a>	获取质量作业详情	300	40
<a href="#">批量删除质量作业</a>	批量删除质量作业。	300	40

## 对账作业接口

表 2-52 对账作业接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取对账作业列表</a>	获取对账作业列表。	300	40
<a href="#">获取对账作业详情</a>	获取对账作业详情。	300	40
<a href="#">批量删除对账作业</a>	批量删除对账作业。	300	40

## 运维管理接口

表 2-53 运维管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取任务执行结果列表</a>	获取任务执行结果列表。	300	40
<a href="#">获取实例结果</a>	获取实例结果。	300	40
<a href="#">数据质量运维管理操作处理&amp;记录</a>	该接口用于数据质量监控实例的问题处理，在控制台上可通过“更多>处理&记录”进入问题处理界面。	300	40

## 任务实例接口

表 2-54 任务实例接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">批量停止实例</a>	批量停止作业实例。	300	40

## 质量报告接口

表 2-55 质量报告接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取质量报告评分体系</a>	获取质量报告评分体系。	300	40
<a href="#">获取质量报告概览</a>	获取质量报告概览。	300	40
<a href="#">获取质量报告趋势</a>	获取质量报告趋势。	300	40
<a href="#">获取质量报告规则评分</a>	获取质量报告规则评分。	300	40
<a href="#">获取质量报告子规则字段评分</a>	获取质量报告子规则字段评分。	300	40
<a href="#">获取质量报告技术报告数据</a>	获取质量报告技术报告数据。	300	40

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取质量报告业务报告数据</a>	获取质量报告业务报告数据。	300	40

## 导入导出接口

表 2-56 导入导出接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">导出资源</a>	该接口用于数据质量导出操作，可通过该接口导出质量作业或者对账作业等属于数据质量服务的资源数据。	1800	20
<a href="#">获取导入导出的任务状态</a>	该接口用于获取导出导入任务的进度状态。	300	40
<a href="#">下载资源任务文件</a>	该接口用于下载资源文件。	300	40
<a href="#">上传导入的文件接口</a>	该接口用于数据质量导入文件使用。	1800	20
<a href="#">导入资源</a>	该接口用于数据质量导入操作使用。	1800	20

## 2.7 数据目录 API 概览

表 2-57 数据目录 API 类型

类型	说明	API流量限制	用户流量限制 (次/s)
<a href="#">业务资产接口</a>	业务资产接口	对应类型的所有API共享 100次/s的API 流量限制	对应类型的所有API共享50 次/s的用户流 量限制
<a href="#">指标资产接口</a>	指标资产接口		
<a href="#">统计资产接口</a>	统计资产接口		
<a href="#">资产管理接口</a>	资产管理接口		
<a href="#">资产分类接口</a>	资产分类接口		
<a href="#">资产分级接口</a>	资产分级		
<a href="#">资产信息</a>	资产信息接口		

类型	说明	API流量限制	用户流量限制 (次/s)
血缘信息	血缘信息接口		
元数据采集任务接口	元数据采集任务接口		
标签接口	标签接口		
数据地图接口	数据地图接口	对应类型的所有API共享100次/s的API流量限制	对应类型的所有API共享50次/s的用户流量限制

## 业务资产接口

表 2-58 业务资产接口

API	说明
查询业务资产	查询业务资产，包含数据规范同步过来的业务对象和逻辑实体。
查询业务资产目录树	逐级查询业务资产目录树，包含数据规范同步过来的业务对象和逻辑实体。

## 指标资产接口

表 2-59 指标资产接口

API	说明
查询指标资产目录树	查询指标资产目录树。
查询指标资产	指标资产查询接口。

## 统计资产接口

表 2-60 统计资产接口

API	说明
获取技术资产统计信息	获取技术资产统计信息。
获取业务资产统计信息	获取业务资产统计信息。

## 资产管理接口

表 2-61 资产管理接口

API	说明
<a href="#">查询技术资产</a>	查询技术资产。
<a href="#">根据guid获取资产详情</a>	根据表guid可以获得表的详情信息，表的详情信息包含column的信息，也可以根据column的guid直接获取column的信息。
<a href="#">添加或修改资产</a>	添加或修改资产。
<a href="#">删除资产</a>	删除资产。

## 资产分类接口

表 2-62 资产分类接口

API	说明
<a href="#">批量资产关联分类</a>	批量资产关联分类：只支持对数据表的列和OBS对象添加分类。
<a href="#">资产关联分类</a>	将一个分类关联到一个或多个指定guid的资产上。
<a href="#">移除资产关联的分类</a>	移除资产关联分类。

## 资产分级接口

表 2-63 资产分级接口

API	说明
<a href="#">批量资产关联密级</a>	批量资产关联密级，单个密级关联到多个资产上。
<a href="#">资产关联密级</a>	关联资产到密级，资产关联指定密级。
<a href="#">移除资产关联密级</a>	移除资产关联密级。

## 资产信息

表 2-64 资产信息

API	说明
<a href="#">查询概要</a>	查询概要。
<a href="#">指定字段采集概要</a>	指定字段采集概要信息接口。

## 血缘信息

表 2-65 血缘信息

API	说明
<a href="#">无血缘关系表查询</a>	无血缘关系表查询，请求参数type_names、classification、type_names、connection_names、query需至少指定一个。
<a href="#">血缘查询</a>	血缘查询。
<a href="#">血缘导入</a>	血缘导入。
<a href="#">创建血缘信息</a>	创建血缘信息

## 元数据采集任务接口

表 2-66 元数据采集任务接口

API	说明
<a href="#">查询采集任务列表</a>	查询采集任务列表。
<a href="#">创建采集任务</a>	创建采集任务。
<a href="#">查询采集任务详情</a>	查询采集任务详情。
<a href="#">编辑采集任务</a>	编辑采集任务。
<a href="#">删除单个采集任务</a>	删除单个采集任务。
<a href="#">启动、调度、停止采集任务</a>	启动、调度、停止采集任务。
<a href="#">获取任务日志</a>	获取任务日志。

## 标签接口

表 2-67 标签接口

API	说明
<a href="#">查询标签列表</a>	查询标签列表。
<a href="#">标签关联到资产</a>	标签关联到资产。

## 数据地图接口

表 2-68 数据地图接口

API	说明
<a href="#">元数据实时同步接口(邀测)</a>	元数据实时同步接口，支持批量。该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">搜索查询标签分页展示(邀测)</a>	搜索查询标签分页展示，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">用户行为分析(邀测)</a>	用户行为分析，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">资产搜索(邀测)</a>	资产搜索，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">创建或修改资产(邀测)</a>	创建或修改资产，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">资产详情(邀测)</a>	资产详情接口，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">资产血缘(邀测)</a>	资产血缘接口，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">批量血缘(邀测)</a>	批量血缘接口，一次性获取所有作业算子的血缘。该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">表关联作业算子列表(邀测)</a>	查询表相关的作业算子列表，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">产出信息(邀测)</a>	查询表相关的作业算子运行实例信息，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。
<a href="#">批量打标签(邀测)</a>	批量给资产打标签。



## 2.8 数据服务 API 概览

表 2-69 数据服务 API 类型

类型	说明
<a href="#">API管理接口</a>	API管理接口。
<a href="#">申请管理接口</a>	申请管理接口。
<a href="#">消息管理接口</a>	消息管理接口。
<a href="#">授权管理接口</a>	授权管理接口。
<a href="#">服务目录管理接口</a>	服务目录管理接口。
<a href="#">网关管理接口</a>	网关管理接口。
<a href="#">应用管理接口</a>	应用管理接口。
<a href="#">总览接口</a>	总览接口。
<a href="#">集群管理接口</a>	集群管理接口。

### API 管理接口

表 2-70 API 管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">创建API</a>	创建API。	2400	120
<a href="#">查询API列表</a>	查询API列表。	2400	120
<a href="#">更新API</a>	更新API。	2400	120
<a href="#">查询API信息</a>	查询API信息。	2400	120
<a href="#">批量删除API</a>	批量删除API。	2400	120
<a href="#">发布API</a>	发布API。API只有发布后，才能够被调用。API发布时，可以将API发送至指定网关。	2400	120
<a href="#">API操作(下线/停用/恢复)</a>	下线API。将已发布的API下线。下线后，所有授权关系都会被解除，API将无法再被调用。	2400	120

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">批量授权API(专享版)</a>	APP创建成功后，还不能访问API，如果想要访问某个API，需要将该API授权给APP。API主动授权成功后，在有效期内，APP即可访问该API。	2400	120
<a href="#">调试API</a>	调试API。	2400	120
<a href="#">API授权操作(授权/取消授权/申请/续约)</a>	API主动授权：API审核人可发起，API主动授权成功后，在有效期内，APP即可访问该API。API授权包含授权和续约两部分功能。	2400	120
<a href="#">查看API发布信息(专享版)</a>	查看API在不同集群上的发布信息。	2400	120
<a href="#">查看API不同操作对应的实例信息(专享版)</a>	查看API不同操作对应的实例信息。	2400	120
<a href="#">查看API调试信息(专享版)</a>	查看API在不同集群上的调试信息。	2400	120
<a href="#">导入包含API信息的EXCEL文件</a>	导入包含API信息的EXCEL文件。	2400	120
<a href="#">导出包含API信息的EXCEL文件</a>	导出包含API信息的EXCEL文件。	2400	120
<a href="#">全量导出包含API的EXCEL压缩文件</a>	全量导出包含API的EXCEL压缩文件。	2400	120
<a href="#">下载EXCEL模板</a>	下载EXCEL模板。	2400	120

## 申请管理接口

表 2-71 申请管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询申请列表</a>	查询申请列表。	2400	120
<a href="#">审核申请</a>	审核申请。	2400	120
<a href="#">获取申请详情</a>	获取申请详情。	2400	120

## 消息管理接口

表 2-72 消息管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询消息列表</a>	查询审核中心的通知消息列表。与申请不同，通知类消息，无法驳回，仅能在指定的时间范围内作出处理。	2400	120
<a href="#">处理消息</a>	对收到的通知消息进行确认，可以在指定的时间范围内选择何时进行处理。	2400	120
<a href="#">获取消息详情</a>	获取消息详情。此功能仅用作信息详情展示，不用做业务处理，因此不展示编号等后台参数。	2400	120

## 授权管理接口

表 2-73 授权管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询API已授权的APP</a>	查询API已授权的APP。	2400	120
<a href="#">查询APP已拥有授权的API</a>	查询APP已拥有授权的API。	2400	120

## 服务目录管理接口

表 2-74 服务目录管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取当前目录下的所有类型列表</a>	获取当前目录下所有类型列表（包括API和目录，均以目录的数据格式形式展示）。	2400	120
<a href="#">获取当前目录下的API列表</a>	获取当前目录下的API列表。	2400	120

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取当前目录下的目录列表(全量)</a>	获取当前目录下的目录列表（全量数据，不分页，推荐仅用于生成目录树等无法分页的场景）。	2400	120
<a href="#">更新服务目录</a>	更新服务目录。	2400	120
<a href="#">查询服务目录</a>	查询服务目录。	2400	120
<a href="#">创建服务目录</a>	创建服务目录。根目录编号为0。	2400	120
<a href="#">批量删除目录</a>	批量删除服务目录。	2400	120
<a href="#">移动当前目录至新目录</a>	移动当前目录至新目录。	2400	120
<a href="#">批量移动API至新目录</a>	批量移动API至新目录。	2400	120
<a href="#">通过路径获取ID</a>	通过路径获取ID。	2400	120
<a href="#">通过ID获取路径</a>	通过ID获取路径。	2400	120
<a href="#">通过目录ID获取路径对象</a>	通过目录id获取路径对象。	2400	120
<a href="#">查询服务目录API列表</a>	查询服务目录API列表。	2400	120

## 网关管理接口

表 2-75 网关管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取网关实例(专享版)</a>	获取网关实例(专享版)。	2400	120
<a href="#">获取网关分组</a>	获取网关分组。	2400	120

## 应用管理接口

表 2-76 应用管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">创建应用</a>	创建应用。	2400	120
<a href="#">查询应用列表</a>	查询应用列表。	2400	120
<a href="#">更新应用</a>	更新应用。	2400	120
<a href="#">删除应用</a>	删除应用。	2400	120
<a href="#">查询应用详情</a>	查询应用详情。	2400	120

## 总览接口

表 2-77 总览接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询统计用户相关的总览开发指标</a>	查询统计用户相关的总览开发指标。	2400	120
<a href="#">查询统计用户相关的总览调用指标</a>	查询统计用户相关的总览调用指标。	2400	120
<a href="#">查询API服务调用TopN</a>	查询API服务调用TopN。	2400	120
<a href="#">查询APP服务使用TopN</a>	查询APP服务使用TopN。	2400	120
<a href="#">查询API统计数据详情</a>	查询API统计数据详情。	2400	120
<a href="#">查询APP统计数据详情</a>	查询APP统计数据详情。	2400	120
<a href="#">查询API仪表盘数据详情</a>	查询API仪表盘数据详情。	2400	120
<a href="#">查询指定API仪表盘数据详情</a>	查询指定API仪表盘数据详情。	2400	120
<a href="#">查询APP仪表盘数据详情</a>	查询APP仪表盘数据详情。	2400	120

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询指定API应用调用TopN</a>	查询指定API应用调用TopN。	2400	120

## 集群管理接口

表 2-78 集群管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询集群概览信息列表</a>	查询集群概览信息列表。	2400	120
<a href="#">查询集群详情信息列表</a>	查询集群详情信息列表。	2400	120
<a href="#">查询集群详情信息</a>	查询集群详情信息。	2400	120
<a href="#">查询数据服务集群访问日志列表</a>	查询数据服务集群访问日志列表。	2400	120
<a href="#">开启数据服务集群OBS日志转储</a>	开启数据服务集群OBS日志转储。	2400	120
<a href="#">开启数据服务集群LTS日志转储</a>	开启数据服务集群LTS日志转储。	2400	120

## 2.9 数据安全 API 概览

表 2-79 数据安全 API 类型

类型	说明
<a href="#">权限管理接口</a>	权限管理接口。
<a href="#">识别规则接口</a>	识别规则接口。
<a href="#">规则分组接口</a>	规则分组接口。
<a href="#">数据权限查询接口</a>	数据权限查询接口。
<a href="#">数据密级接口</a>	数据密级接口。
<a href="#">权限应用接口</a>	权限应用接口。
<a href="#">敏感数据分布接口</a>	敏感数据分布接口。

类型	说明
用户同步接口	用户同步接口。
队列权限接口	队列权限接口。
数据分类接口	数据分类接口。
数据安全诊断接口	数据安全诊断接口。
空间资源权限策略管理接口	空间资源权限策略管理接口。
安全管理员接口	安全管理员接口。
动态数据脱敏接口	动态数据脱敏接口。
权限审批接口	权限审批接口。
权限申请接口	权限申请接口。

## 权限管理接口

表 2-80 权限管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
创建权限集	创建权限集。	1800	20
查询权限集列表	查询权限集列表。	3600	40
查询权限集	查询权限集。	3600	40
删除权限集	删除权限集。	1800	20
更新权限集	更新权限集。	1800	20
添加权限集成员	添加权限集成员。	1800	20
查询权限集成员列表	查询权限集成员列表。	3600	40
批量删除权限集成员	批量删除权限集成员。	1800	20
添加权限集的权限	添加权限集的权限。	1800	20
查询权限集的权限列表	查询权限集的权限列表。	3600	40
删除权限集的权限	删除权限集的权限。	1800	20
更新权限集的权限	更新权限集的权限。	1800	20

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询数据源可配置权限</a>	查询数据源可配置权限。	3600	40
<a href="#">查询权限集中配置的url信息</a>	查询权限集中配置的url信息，例如obs、hdfs等组件中的url路径。	3600	40
<a href="#">查询数据操作信息</a>	查询数据操作信息。	3600	40

## 识别规则接口

表 2-81 识别规则接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询识别规则列表</a>	查询识别规则列表。	3600	40
<a href="#">创建识别规则</a>	创建识别规则。	1800	20
<a href="#">查询特定识别规则</a>	查询特定识别规则。	3600	40
<a href="#">删除识别规则</a>	删除识别规则。	1800	20
<a href="#">修改识别规则接口</a>	修改识别规则接口。	1800	20
<a href="#">批量删除识别规则接口</a>	批量删除识别规则接口。	1800	20
<a href="#">修改识别规则状态接口</a>	修改识别规则状态接口。	1800	20
<a href="#">创建组合识别规则</a>	创建组合识别规则。	1800	20
<a href="#">修改组合识别规则</a>	修改组合识别规则。	1800	20
<a href="#">组合识别规则测试</a>	组合识别规则测试。	1800	20

## 规则分组接口

表 2-82 规则分组接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询规则组列表</a>	查询规则组列表。	3600	40



API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">创建规则分组接口</a>	创建规则分组接口	1800	20
<a href="#">查询规则组</a>	查询规则组。	3600	40
<a href="#">修改规则分组接口</a>	修改规则分组接口	1800	20
<a href="#">删除规则分组接口</a>	删除规则分组接口	1800	20

## 数据权限查询接口

表 2-83 数据权限查询接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询角色对一组库、表的可配置操作权限信息</a>	查询角色对一组库、表的可配置操作权限信息。	3600	40

## 数据密级接口

表 2-84 数据密级接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取数据密级</a>	获取数据密级。	3600	40
<a href="#">创建数据密级</a>	创建数据密级。	1800	20
<a href="#">根据指定的id查询数据密级</a>	根据指定的id查询数据密级。	3600	40
<a href="#">删除指定的id的数据密级</a>	删除指定的id的数据密级。	1800	20
<a href="#">根据指定的id修改数据密级</a>	根据指定的id修改数据密级。	1800	20
<a href="#">批量删除数据密级</a>	批量删除数据密级。	1800	20

## 权限应用接口

表 2-85 权限应用接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询数据开发细粒度连接列表</a>	查询数据开发细粒度连接列表 (全量)。	3600	40
<a href="#">批量更新数据开发连接细粒度认证状态</a>	批量更新数据开发连接细粒度认证状态。	1800	20
<a href="#">测试数据开发连接细粒度连通性</a>	测试数据开发连接细粒度连通性。	1800	20

## 敏感数据分布接口

表 2-86 敏感数据分布接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询敏感数据发现概览结果(以分类和密级为单位)</a>	查询敏感数据发现概览结果(以分类和密级为单位)。	3600	40

## 用户同步接口

表 2-87 用户同步接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询用户同步列表</a>	查询用户同步列表。	3600	40
<a href="#">查询单个用户同步任务</a>	查询单个用户同步任务。	3600	40

## 队列权限接口

表 2-88 队列权限接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询当前空间下分配的队列资源</a>	查询当前空间下分配的队列资源。	3600	40
<a href="#">分配队列资源给指定空间</a>	分配队列资源给指定空间。	1800	20
<a href="#">修改当前空间下分配的队列资源</a>	修改当前空间下分配的队列资源。	1800	20
<a href="#">删除当前空间下分配的队列资源</a>	删除当前空间下分配的队列资源。	1800	20

## 数据分类接口

表 2-89 数据分类接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">导入预置分类</a>	导入预置分类。	1800	20
<a href="#">查询数据分类列表</a>	查询数据分类列表。	3600	40

## 数据安全诊断接口

表 2-90 数据安全诊断接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">执行数据安全诊断</a>	执行数据安全诊断。	1800	20
<a href="#">查询不合理的权限配置</a>	查询不合理的权限配置。	3600	40
<a href="#">查询数据权限控制模块诊断结果</a>	查询数据权限控制模块诊断结果。	3600	40
<a href="#">查询敏感数据保护模块诊断结果</a>	查询敏感数据保护模块诊断结果	3600	40

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询未进行静态脱敏任务的表信息</a>	查询未进行静态脱敏任务的表信息	3600	40
<a href="#">查询数据源防护模块诊断结果</a>	查询数据源防护模块诊断结果	3600	40

## 空间资源权限策略管理接口

表 2-91 空间资源权限策略管理接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询空间资源权限策略列表</a>	查询空间资源权限策略列表	3600	40
<a href="#">创建空间资源权限策略</a>	创建空间资源权限策略	1800	20
<a href="#">查询单个资源权限策略</a>	查询单个资源权限策略	3600	40
<a href="#">更新空间资源权限策略</a>	更新空间资源权限策略	1800	20
<a href="#">批量删除资源权限策略</a>	批量删除资源权限策略	1800	20

## 安全管理员接口

表 2-92 安全管理员接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查看安全管理员</a>	查看安全管理员。	3600	40
<a href="#">创建或更新安全管理员</a>	创建或更新安全管理员。	1800	20

## 动态数据脱敏接口

表 2-93 动态数据脱敏接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">查询动态数据脱敏策略列表</a>	查询动态数据脱敏策略列表。	3600	40
<a href="#">创建数据脱敏策略</a>	创建动态数据脱敏策略。	1800	20
<a href="#">查询某个脱敏策略的详细信息</a>	查询某个脱敏策略的详细信息。	3600	40
<a href="#">更新动态数据脱敏策略</a>	更新动态数据脱敏策略。	1800	20
<a href="#">批量删除动态脱敏策略</a>	批量删除动态脱敏策略。	1800	20

## 权限审批接口

表 2-94 权限审批接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">获取表计权限审批人列表</a>	获取表计权限审批人列表	3600	40
<a href="#">提交表计权限申请</a>	提交表计权限申请	1800	20
<a href="#">审批通过工单</a>	审批通过工单。	1800	20
<a href="#">驳回工单</a>	驳回工单。	1800	20
<a href="#">获取审批工单列表</a>	获取审批工单列表	3600	40

## 权限申请接口

表 2-95 权限申请接口

API	说明	API流量限制 (次/10s)	用户流量限制 (次/10s)
<a href="#">撤回工单申请</a>	撤回工单申请。	1800	20

# 3 如何调用 API

## 3.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的**管理员创建IAM用户**接口为例说明如何调用API。该API用于管理员创建IAM用户。

您还可以通过这个视频教程了解如何构造请求调用API：<https://bbs.huaweicloud.com/videos/102987>。

### 请求 URI

请求URI由如下部分组成：

**{URI-scheme}://{Endpoint}/{resource-path}?{query-string}**

尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

表 3-1 URI 中的参数说明

参数	描述
URI-scheme	表示用于传输请求的协议，当前所有API均采用HTTPS协议。
Endpoint	指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同。 终端节点（Endpoint）即调用API的 <b>请求地址</b> ，不同服务不同区域的终端节点不同。本服务的Endpoint您可以从 <b>终端节点Endpoint</b> 获取。 例如IAM服务在“华北-北京四”区域的Endpoint为“iam.cn-north-4.myhuaweicloud.com”。
resource-path	资源路径，也即API访问路径。从具体API的URI模块获取，例如“管理员创建IAM用户”接口的resource-path为“/v3.0/OS-USER/users”。

参数	描述
query-string	查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

例如您需要创建IAM用户，由于IAM为全局服务，则使用任一区域的Endpoint（比如“华北-北京四”区域的Endpoint：iam.cn-north-4.myhuaweicloud.com），并在[管理员创建IAM用户](#)接口的URI部分找到resource-path（/v3.0/OS-USER/users），拼接起来如下所示。

`https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users`

图 3-1 URI 示意图



### 说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

## 请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

表 3-2 HTTP 方法

方法	说明
GET	请求服务器返回指定资源。
PUT	请求服务器更新指定资源。
POST	请求服务器新增资源或执行特殊操作。
DELETE	请求服务器删除指定资源，如删除对象等。
HEAD	请求服务器资源头部。
PATCH	请求服务器更新资源的部分内容。 当资源不存在的时候，PATCH可能会去创建一个新的资源。

在IAM的[管理员创建IAM用户](#)接口的URI部分，您可以看到其请求方法为“POST”，则其请求为：

`POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users`

## 请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

详细的公共请求消息头字段请参见[表3-3](#)。

表 3-3 公共请求消息头

名称	描述	是否必选	示例
Content-Type	消息体的类型（格式），默认取值为“application/json”。 如果请求消息体中含有中文字符，则还需要通过charset=utf8指定中文字符集。	是	application/json;charset=utf8
Content-Length	请求body长度，单位为Byte。	否	3495
X-Language	请求语言。	否	en_us
X-Auth-Token	用户Token。用户Token是调用 <a href="#">获取用户Token</a> 接口的响应值，因此调用 <a href="#">获取用户Token</a> 接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。	使用Token认证时该字段必选。	注：以下仅为Token示例片段 MIIPAgYJKoZlhvc NAQcCo...ggg1BBI INPXsidG9rZ
Authorization	签名认证信息。 当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。	使用AK/SK认证时必选。	-
X-Sdk-Date	请求的发生时间，格式为(YYYYMMDD'T'HHMMSS'Z')，取值为当前系统的GMT时间。 当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。	使用AK/SK认证时必选。	20150907T101459Z
Host	请求的服务器信息，从服务API的URL中获取。值为hostname[:port]。端口缺省时使用默认的端口，https的默认端口为443。	使用AK/SK认证时必选。	code.test.com or code.test.com:443



名称	描述	是否必选	示例
X-Project-Id	子项目ID。请参考 <a href="#">项目ID和账号ID</a> 章节获取项目ID。	调用在子项目中的服务接口，并使用AK/SK认证时必选。	e9993fc787d94b6c886cbaa340f9c0f4
X-Domain-ID	账号ID。请参考 <a href="#">项目ID和账号ID</a> 章节获取账号ID。	调用全局服务接口，并使用AK/SK认证时必选。	0a2f5e28af00f3850f31c00f31e884e0
workspace	工作空间id，获取方式请参见 <a href="#">实例ID和工作空间ID</a> 。	<ul style="list-style-type: none"> <li>调用批量数据迁移或实时数据接入API时无需配置。</li> <li>调用数据开发API时，有多个DataArts Studio实例时必选。只有一个DataArts实例时可选。不填写则默认查询default工作空间数据，如需查询其他工作空间需要带上该消息头。</li> <li>调用其他组件API时必选，否则会调用失败。</li> </ul>	d1cd7861478748a6925bc02f47c69279
X-Dlm-Type	数据服务版本类型，用于请求数据服务时，指定共享版/专享版	调用数据服务接口时必选	共享版: SHARED 专享版: EXCLUSIVE

### 📖 说明

API同时支持使用AK/SK认证，AK/SK认证是使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。

AK/SK认证的详细说明请参见[认证鉴权](#)的“AK/SK认证”。

对于[管理员创建IAM用户](#)接口，使用AK/SK方式认证时，添加消息头后的请求如下所示。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

## 请求消息体

请求消息体通常以结构化格式（如JSON或XML）发出，与请求消息头中Content-Type对应，传递除请求消息头之外的内容。若请求消息体中的参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

对于**管理员创建IAM用户**接口，您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示，加粗的斜体字段需要根据实际值填写：

- **accountid**为IAM用户所属的账号ID。
- **username**为要创建的IAM用户名。
- **email**为IAM用户的邮箱。
- **\*\*\*\*\***为IAM用户的登录密码。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

```
{
  "user": {
    "domain_id": "accountid",
    "name": "username",
    "password": "*****",
    "email": "email",
    "description": "IAM User Description"
  }
}
```

到这里为止，一个API请求所需要的内容已经准备完成，您可以使用curl、Postman或直接编写代码等方式发送请求调用API。

## 3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证通用请求。
- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。

### 📖 说明

仅当创建IAM用户时的访问方式勾选“编程访问”后，此IAM用户才能通过认证鉴权，从而使用API、SDK等方式访问DataArts Studio。

## Token 认证

### 📖 说明

- Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。
- 使用Token前请确保Token离过期有足够的时间，防止调用API的过程中Token过期导致调用API失败。

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

Token可通过调用**获取用户Token**接口获取，调用本服务API需要project级别的Token，即调用**获取用户Token**接口时，请求body中auth.scope的取值需要选择project，如下所示。

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****#",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "id": "xxxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

获取Token后，再调用其他接口时（以数据开发组件的“查询连接列表”接口为例），您需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFJ....”，则调用接口时将“X-Auth-Token: ABCDEFJ....”加到请求消息头即可，如下所示。

```
GET https://iam.cn-north-4.myhuaweicloud.com/v1/{project_id}/connections
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

您还可以通过这个视频教程了解如何使用Token认证：<https://bbs.huaweicloud.com/videos/101333>。

## AK/SK 认证

### 📖 说明

- AK/SK签名认证方式仅支持消息体大小12MB以内，12MB以上的请求请使用Token认证。
- AK/SK既可以使用永久访问密钥中的AK/SK，也可以使用临时访问密钥中的AK/SK，但使用临时访问密钥的AK/SK时需要额外携带“X-Security-Token”字段，字段值为临时访问密钥的security\_token。

AK/SK认证就是使用AK/SK对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- AK(Access Key ID)：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- SK(Secret Access Key)：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

您可以通过如下方式获取访问密钥。

1. 登录控制台，在用户名下拉列表中选择“我的凭证”。
2. 进入“我的凭证”页面，选择“访问密钥 > 新增访问密钥”，如图3-2所示。

图 3-2 单击新增访问密钥



3. 单击“确定”，根据浏览器提示，保存密钥文件。密钥文件会直接保存到浏览器默认的下载文件夹中。打开名称为“credentials.csv”的文件，即可查看访问密钥（Access Key Id和Secret Access Key）。

#### 说明

- 每个用户仅允许新增两个访问密钥。
- 为保证访问密钥的安全，访问密钥仅在初次生成时自动下载，后续不可再次通过管理控制台界面获取。请在生成后妥善保管。

使用AK/SK认证时，您可以基于签名算法使用AK/SK对请求进行签名，也可以使用专门的签名SDK对请求进行签名。详细的签名方法和SDK使用方法请参见[API签名指南](#)。

#### 须知

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

您也可以通过这个视频教程了解AK/SK认证的使用：<https://bbs.huaweicloud.com/videos/100697>。

## 3.3 返回结果

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

### 状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见[状态码](#)。

对于[获取用户Token](#)接口，如果调用后返回状态码为“201”，则表示请求成功。

### 响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于[管理员创建IAM用户](#)接口，如果调用后返回状态码为“201”，则表示请求成功。

### 响应消息体

响应消息体通常以结构化格式（如JSON或XML）返回，与响应消息头中Content-Type对应，传递除响应消息头之外的内容。

对于**管理员创建IAM用户**接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "user": {
    "id": "c131886aec...",
    "name": "IAMUser",
    "description": "IAM User Description",
    "areacode": "",
    "phone": "",
    "email": "***@***.com",
    "status": null,
    "enabled": true,
    "pwd_status": false,
    "access_mode": "default",
    "is_domain_owner": false,
    "xuser_id": "",
    "xuser_type": "",
    "password_expires_at": null,
    "create_time": "2024-05-21T09:03:41.000000",
    "domain_id": "d78cbac1.....",
    "xdomain_id": "30086000.....",
    "xdomain_type": "",
    "default_project_id": null
  }
}
```

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{
  "error_msg": "Request body is invalid.",
  "error_code": "IAM.0011"
}
```

其中，error\_code表示错误码，error\_msg表示错误描述信息。

# 4 数据集成 API

## 4.1 集群管理

### 4.1.1 查询集群详情

#### 功能介绍

查询集群详情接口。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1.1/{project\_id}/clusters/{cluster\_id}

表 4-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。

## 请求参数

表 4-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 200

表 4-3 响应 Body 参数

参数	参数类型	描述
publicEndpoint	String	集群绑定的EIP。
instances	Array of <b>ClusterDetail Instance</b> objects	集群的节点信息，请参见instances参数说明。
security_group_id	String	安全组id。
subnet_id	String	子网id。
vpc_id	String	虚拟私有云ID。
customerConfig	<b>CustomerConfig</b> object	用户配置信息。
datastore	<b>Datastore</b> object	cdm信息。
isAutoOff	Boolean	自动关机。
publicEndpointDomainName	String	集群绑定的EIP域名。
bakExpectedStartTime	String	开始时间。
bakKeepDay	Integer	保留时间。
maintainWindow	<b>maintainWindow</b> object	维护窗口。

参数	参数类型	描述
recentEvent	Integer	事件数。
flavorName	String	规格名称。
azName	String	az名称。
endpointDomainName	String	对端域名。
publicEndpointStatus	<b>publicEndpointStatus</b> object	EIP状态信息。
isScheduleBootOff	Boolean	选择是否启用定时开关机功能。定时开关机功能和自动关机功能不可同时开启。
namespace	String	命名空间。
eipId	String	弹性ip id。
failedReasons	<b>FailedReasons</b> object	失败原因。集群处于正常状态时不返回。
dbuser	String	数据库用户。
links	Array of <b>ClusterLinks</b> objects	集群连接信息。
clusterMode	String	集群模式：sharding(分片集群)。
task	<b>ClusterTask</b> object	任务信息。
created	String	集群创建时间，格式为ISO8601：YYYY-MM-DDThh:mm:ssZ。
statusDetail	String	集群状态描述：Normal（正常）。
config_status	String	集群配置状态： <ul style="list-style-type: none"> <li>● In-Sync：配置已同步。</li> <li>● Applying：配置中。</li> <li>● Sync-Failure：配置失败。</li> </ul>
actionProgress	<b>ActionProgress</b> object	集群操作进度，任务信息，由key、value组成。key值为正在进行的任务，value值为正在进行任务的进度。示例如 "action_progress": {"SNAPSHOTTING": "16%"}
name	String	集群名称。
id	String	集群ID。
isFrozen	String	集群是否冻结：0：否，1：是。



参数	参数类型	描述
actions	Array of strings	集群配置状态：In-Sync：配置已同步。 Applying：配置中。Sync-Failure：配置失败。
updated	String	集群更新时间，格式为 ISO8601：YYYY-MM-DDThh:mm:ssZ。
status	String	集群状态： <ul style="list-style-type: none"> <li>• 100：创建中。</li> <li>• 200：正常。</li> <li>• 300：失败。</li> <li>• 303：创建失败。</li> <li>• 800：冻结。</li> <li>• 900：已关机。</li> <li>• 910：正在关机。</li> <li>• 920：正在开机。</li> </ul>

表 4-4 ClusterDetailInstance

参数	参数类型	描述
flavor	<b>flavor</b> object	节点的虚拟机规格，请参见flavor参数说明（查询集群列表时返回值为null）。
volume	<b>volume</b> object	节点的磁盘信息，请参见volume参数说明（查询集群列表时返回值为null）。
status	String	节点状态： <ul style="list-style-type: none"> <li>• 100：创建中。</li> <li>• 200：正常。</li> <li>• 300：失败。</li> <li>• 303：创建失败。</li> <li>• 400：已删除。</li> <li>• 800：冻结。</li> </ul>
actions	Array of strings	节点操作状态列表： <ul style="list-style-type: none"> <li>• REBOOTING：重启中。</li> <li>• RESTORING：恢复中。</li> <li>• REBOOT_FAILURE：重启失败。</li> </ul>
type	String	节点类型，只支持一种类型“cdm”。
id	String	节点的虚拟机ID。
name	String	节点的虚拟机名称。

参数	参数类型	描述
isFrozen	String	节点是否冻结: 0: 否。1: 是。
components	String	组件。
config_status	String	节点配置状态 ( 查询集群列表时为null ) : <ul style="list-style-type: none"> <li>• In-Sync: 配置已同步。</li> <li>• Applying: 配置中。</li> <li>• Sync-Failure: 配置失败。</li> </ul>
role	String	实例角色。
group	String	分组。
links	Array of <b>ClusterLinks</b> objects	链接信息 ( 查询集群列表时返回值为null ) 。
paramsGroupid	String	组件分组id。
publicip	String	公网ip。
managelp	String	管理ip。
trafficip	String	流量ip。
shard_id	String	分片id。
manage_fix_ip	String	管理修复ip。
private_ip	String	私有ip。
internal_ip	String	内部ip。
resource	Array of <b>Resource</b> objects	资源信息 ( 查询集群列表时返回值为null ) 。

表 4-5 flavor

参数	参数类型	描述
id	String	节点虚拟机的规格ID。
links	Array of <b>ClusterLinks</b> objects	链接信息。

表 4-6 volume

参数	参数类型	描述
type	String	节点的磁盘类型，只支持本地磁盘。
size	Long	节点磁盘大小，单位G。

表 4-7 Resource

参数	参数类型	描述
resource_id	String	资源id。
resource_type	String	资源类型：server(服务器)。

表 4-8 CustomerConfig

参数	参数类型	描述
failureRemind	String	失败提醒。
clusterName	String	集群类型。
serviceProvider	String	服务提供。
localDisk	String	是否本地磁盘。
ssl	String	是否使用ssl。
createFrom	String	创建来源。
resourceId	String	资源ID。
flavorType	String	规格类型。
workSpaceId	String	工作空间ID。
trial	String	适用。

表 4-9 Datastore

参数	参数类型	描述
type	String	类型，一般为cdm。
version	String	集群版本。

表 4-10 maintainWindow

参数	参数类型	描述
day	String	周几。
startTime	String	开始时间。
endTime	String	结束时间。

表 4-11 publicEndpointStatus

参数	参数类型	描述
status	String	状态。
errorMessage	String	错误信息。

表 4-12 FailedReasons

参数	参数类型	描述
CREATE_FAILED	<b>CREATE_FAILED</b> object	集群创建失败原因。

表 4-13 CREATE\_FAILED

参数	参数类型	描述
errorCode	String	错误码。
errorMsg	String	失败原因。

表 4-14 ClusterLinks

参数	参数类型	描述
rel	String	关系。
href	String	链接地址。

表 4-15 ClusterTask

参数	参数类型	描述
description	String	任务描述。

参数	参数类型	描述
id	String	任务id。
name	String	任务名称。

表 4-16 ActionProgress

参数	参数类型	描述
CREATING	String	创建集群进度，例如：29%。
GROWING	String	扩容集群进度，例如：29%。
RESTORING	String	恢复集群进度，例如：29%。
SNAPSHOTTING	String	集群快照进度，例如：29%。
REPAIRING	String	修复集群进度，例如：29%。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/bae65496-643e-47ca-84af-948672de7eeb
```

## 响应示例

状态码： 200

OK。

```
{
  "publicEndpoint": "49.xx.xx.10",
  "instances": [ {
    "flavor": {
      "id": "fb8fe666-6734-4b11-bc6c-43d11db3c745"
    },
    "volume": {
      "size": "100",
      "type": "LOCAL_DISK"
    },
    "name": "cdm-c018",
    "id": "635dce67-3df8-4756-b4c7-90e45e687367",
    "isFrozen": "0",
    "type": "cdm",
    "actions": "REBOOTING",
    "config_status": "In-Sync",
    "status": "200"
  } ],
  "created": "2018-09-05T08:38:25",
  "statusDetail": "Normal",
  "actionProgress": { },
  "name": "cdm-c018",
  "id": "bae65496-643e-47ca-84af-948672de7eeb",
  "isFrozen": "0",
  "actions": "REBOOTING",
  "updated": "2018-09-05T08:38:25",
  "status": "200"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ShowClusterDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowClusterDetailRequest request = new ShowClusterDetailRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowClusterDetailResponse response = client.showClusterDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowClusterDetailRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_cluster_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowClusterDetailRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowClusterDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.2 删除集群

#### 功能介绍

删除集群接口。

#### 调用方法

请参见[如何调用API](#)。

#### URI

DELETE /v1.1/{project\_id}/clusters/{cluster\_id}

表 4-17 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。



## 请求参数

表 4-18 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-19 请求 Body 参数

参数	是否必选	参数类型	描述
keep_last_manual_backup	是	Integer	日志备份数，填写为默认填0即可。

## 响应参数

状态码： 202

表 4-20 响应 Body 参数

参数	参数类型	描述
jobId	String	作业ID。

## 请求示例

```
DELETE /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920
{
  "keep_last_manual_backup": 0
}
```

## 响应示例

状态码： 202

Accepted。

```
{
  "jobId": "ff8080815e55125a015e552eddba001a"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class DeleteClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteClusterRequest request = new DeleteClusterRequest();
        request.withClusterId("{cluster_id}");
        CdmDeleteClusterReq body = new CdmDeleteClusterReq();
        body.withKeepLastManualBackup(0);
        request.withBody(body);
        try {
            DeleteClusterResponse response = client.deleteCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = CdmClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CdmRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteClusterRequest()
    request.cluster_id = "{cluster_id}"
    request.body = CdmDeleteClusterReq(
        keep_last_manual_backup=0
    )
    response = client.delete_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteClusterRequest{}
    request.ClusterId = "{cluster_id}"
    request.Body = &model.CdmDeleteClusterReq{
        KeepLastManualBackup: int32(0),
    }
    response, err := client.DeleteCluster(request)
```

```
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
202	Accepted。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.3 查询所有可用区

#### 功能介绍

查询CDM集群的所有可用区。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1.1/{project\_id}/regions/{region\_id}/availability\_zones

表 4-21 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

参数	是否必选	参数类型	描述
region_id	是	String	通过调用IAM服务的“查询区域列表”接口获取响应消息中的区域ID。

## 请求参数

表 4-22 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 4-23 响应 Body 参数

参数	参数类型	描述
regionId	String	区域ID。
defaultAZ	String	默认可用区。
availableZones	Array of CdmClusterAvailabilityZone objects	可用区。

表 4-24 CdmClusterAvailabilityZone

参数	参数类型	描述
availableZoneId	String	可用区ID。
availableZoneName	String	可用区名称。
availableZoneCode	String	可用区码。
azStatus	String	可用区状态。

参数	参数类型	描述
type	String	可用区类型。
tags	Object	可用区tag。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/regions/xxx-xxx-xxx/availability_zones
```

## 响应示例

**状态码： 200**

请求成功。

```
{
  "regionId": "xxx-xxx-xxx",
  "defaultAZ": "xxx-xxx-xxx",
  "availableZones": [ {
    "availableZoneId": "xxx-xxx-xxx",
    "availableZoneName": "xxx-xxx-xxx",
    "availableZoneCode": "xxx-xxx-xxx",
    "azStatus": "Available",
    "type": null,
    "tags": null
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ShowAvailabilityZonesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
```

```

        .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAvailabilityZonesRequest request = new ShowAvailabilityZonesRequest();
        request.withRegionId("{region_id}");
        try {
            ShowAvailabilityZonesResponse response = client.showAvailabilityZones(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}

```

## Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowAvailabilityZonesRequest()
        request.region_id = "{region_id}"
        response = client.show_availability_zones(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"

```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAvailabilityZonesRequest{}
    request.RegionId = "{region_id}"
    response, err := client.ShowAvailabilityZones(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。



## 错误码

请参见[错误码](#)。

## 4.1.4 查询支持的版本

### 功能介绍

查询CDM集群支持的版本。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1.1/{project\_id}/datastores

表 4-25 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 4-26 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

### 响应参数

状态码： 200

表 4-27 响应 Body 参数

参数	参数类型	描述
datastores	Array of <a href="#">CdmClusterDatastore</a> objects	数据库列表。

表 4-28 CdmClusterDatastore

参数	参数类型	描述
id	String	服务ID，用于区分不同服务。
name	String	服务名称。
bigclusterEnable	Boolean	是否支持大规格集群。
defaultVersion	String	默认版本。
versions	Array of <b>CdmClusterVersion</b> objects	版本。

表 4-29 CdmClusterVersion

参数	参数类型	描述
active	String	版本状态。
id	String	版本ID。
image	String	版本镜像。
name	String	版本名称。
packages	String	版本的包。
datastore	String	服务ID，用于区分不同服务。
links	Array of <b>ClusterLinks</b> objects	链接信息。

表 4-30 ClusterLinks

参数	参数类型	描述
rel	String	关系。
href	String	链接地址。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/datastores
```

## 响应示例

状态码： 200

请求成功。

```
[ {
  "id" : "736270b9-27c7-4f03-823b-447d8245e1c2",
  "name" : "cdm",
  "bigclusterEnable" : false,
  "defaultVersion" : "2.9.3.300",
  "links" : null,
  "versions" : [ {
    "active" : "1",
    "id" : "e8a8b8cc-63f8-4fb5-8d4a-24c502317b11",
    "image" : null,
    "links" : [ {
      "rel" : "self",
      "href" : "https://10.63.25.93:443/rds/v1.0/datastores/736270b9-27c7-4f03-823b-447d8245e1c2"
    } ],
    "rel" : "bookmark",
    "href" : "https://10.63.25.93:443/datastores/736270b9-27c7-4f03-823b-447d8245e1c2"
  } ],
  "name" : "2.9.3.300",
  "packages" : "cdm",
  "datastore" : "736270b9-27c7-4f03-823b-447d8245e1c2"
} ]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ShowDatastoresSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDatastoresRequest request = new ShowDatastoresRequest();
        try {
            ShowDatastoresResponse response = client.showDatastores(request);
            System.out.println(response.toString());
        }
    }
}
```

```
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowDatastoresRequest()
        response = client.show_datastores(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cdm.NewCdmClient(
    cdm.CdmClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowDatastoresRequest{}
response, err := client.ShowDatastores(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.5 查询版本规格

#### 功能介绍

按版本ID查询所有兼容规格。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1.1/{project\_id}/datastores/{datastore\_id}/flavors

表 4-31 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
datastore_id	是	String	版本ID，获取方法请参见 <a href="#">CDM支持的版本</a> 。

## 请求参数

表 4-32 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 4-33 响应 Body 参数

参数	参数类型	描述
id	String	服务ID，用于区分不同服务。
dbname	String	db名称，一般为cdm。
versions	Array of <a href="#">CdmClusterDatastoreVersion</a> objects	版本信息列表。

表 4-34 CdmClusterDatastoreVersion

参数	参数类型	描述
id	String	版本ID。
name	String	版本名称。
flavors	Array of <b>CdmClusterFlavor</b> objects	规格信息。

表 4-35 CdmClusterFlavor

参数	参数类型	描述
cpu	Integer	CPU。
ram	Integer	内存。
name	String	规格名称。
region	String	region。
typename	String	类型名称。
clusterMode	String	集群模式。
status	String	规格状态。
str_id	String	规格ID。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/datastores/736270b9-27c7-4f03-823b-447d8245e1c2/flavors
```

## 响应示例

**状态码： 200**

请求成功。

```
{
  "id": "736270b9-27c7-4f03-823b-447d8245e1c2",
  "dbname": "cdm",
  "versions": [ {
    "id": "e8a8b8cc-63f8-4fb5-8d4a-24c502317b11",
    "name": "2.9.3.300",
    "flavors": [ {
      "cpu": 4,
      "ram": 8,
      "name": "cdm.small",
      "region": "xxx-xxx-xxx",
      "typename": "cdm",
      "clusterMode": "sharding",
      "status": "abandon",
      "str_id": "a79fd5ae-1833-448a-88e8-3ea2b913e1f6"
    }
  ]
}
```

```
    }  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;  
import com.huaweicloud.sdk.cdm.v1.*;  
import com.huaweicloud.sdk.cdm.v1.model.*;  
  
public class ShowFlavorsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CdmClient client = CdmClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowFlavorsRequest request = new ShowFlavorsRequest();  
        request.withDatastoreId("{datastore_id}");  
        try {  
            ShowFlavorsResponse response = client.showFlavors(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os
```



```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowFlavorsRequest()
        request.datastore_id = "{datastore_id}"
        response = client.show_flavors(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowFlavorsRequest{}
    request.DatastoreId = "{datastore_id}"
```

```
response, err := client.ShowFlavors(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.6 查询规格详情

#### 功能介绍

查询指定规格ID的规格详情。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1.1/{project\_id}/flavors/{flavor\_id}

表 4-36 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
flavor_id	是	String	规格ID，获取方法请参见 <a href="#">查询版本规格</a> 。

## 请求参数

表 4-37 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 4-38 响应 Body 参数

参数	参数类型	描述
name	String	规格名称。
str_id	String	规格ID。
flavor_detail	Array of <a href="#">flavorAttribute</a> objects	规格详细列表。

表 4-39 flavorAttribute

参数	参数类型	描述
name	String	规格属性名称，如mem、cpu。
value	String	规格属性值。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/flavors/a79fd5ae-1833-448a-88e8-3ea2b913e1f6
```

## 响应示例

**状态码： 200**

请求成功。

```
{
  "str_id" : "a79fd5ae-1833-448a-88e8-3ea2b913e1f6",
  "name" : "cdm.large",
  "flavor_detail" : [ {
    "name" : "cpu",
    "value" : 8
  }, {
    "name" : "mem",
    "value" : 16
  }, {
    "name" : "volumeType",
    "value" : "SATA"
  }, {
    "name" : "flavor",
    "value" : "s6.2xlarge.2"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ShowFlavorDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowFlavorDetailRequest request = new ShowFlavorDetailRequest();
        request.withFlavorId("{flavor_id}");
        try {
            ShowFlavorDetailResponse response = client.showFlavorDetail(request);
        }
    }
}
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowFlavorDetailRequest()
        request.flavor_id = "{flavor_id}"
        response = client.show_flavor_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cdm.NewCdmClient(
    cdm.CdmClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowFlavorDetailRequest{}
request.FlavorId = "{flavor_id}"
response, err := client.ShowFlavorDetail(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.7 查询所有集群的企业项目 ID

#### 功能介绍

查询当前项目下的所有集群的企业项目ID。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1.1/{project\_id}/enterprise-projects

表 4-40 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 4-41 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 4-42 响应 Body 参数

参数	参数类型	描述
resources	Array of <a href="#">CdmClusterEnterpriseProject</a> objects	集群企业项目列表。

表 4-43 CdmClusterEnterpriseProject

参数	参数类型	描述
cluster_id	String	集群ID。
sys_tags	Array of <a href="#">sys_tags</a> objects	企业项目列表。

表 4-44 sys\_tags

参数	参数类型	描述
value	String	企业项目ID。
key	String	该值目前固定为“_sys_enterprise_project_id”。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/enterprise-projects
```

## 响应示例

**状态码： 200**

请求成功。

```
{
  "resources": [ {
    "cluster_id": "2b2a676f-0b91-4ad5-8c24-ec61be586fd1",
    "sys_tags": [ {
      "key": "_sys_enterprise_project_id",
      "value": "1ce45885-4033-40d2-bdde-d4dbaceb387d"
    } ]
  }, {
    "cluster_id": "387f8b24-c4b0-4211-97fe-745d0e88dc88",
    "sys_tags": [ {
      "key": "_sys_enterprise_project_id",
      "value": 0
    } ]
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ShowEnterpriseProjectsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
    }
}
```



```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

CdmClient client = CdmClient.newBuilder()
    .withCredential(auth)
    .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
    .build();
ShowEnterpriseProjectsRequest request = new ShowEnterpriseProjectsRequest();
try {
    ShowEnterpriseProjectsResponse response = client.showEnterpriseProjects(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowEnterpriseProjectsRequest()
        response = client.show_enterprise_projects(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowEnterpriseProjectsRequest{}
    response, err := client.ShowEnterpriseProjects(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。

状态码	描述
503	服务不可用。

## 错误码

请参见[错误码](#)。

## 4.1.8 查询集群的企业项目 ID

### 功能介绍

查询指定集群的企业项目ID。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1.1/{project\_id}/clusters/{cluster\_id}/enterprise-projects

表 4-45 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	CDM集群ID，获取方法请参见 <a href="#">查询集群列表</a> 。

### 请求参数

表 4-46 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

### 响应参数

状态码： 200

表 4-47 响应 Body 参数

参数	参数类型	描述
sys_tags	Array of <a href="#">sys_tags</a> objects	企业项目列表。

表 4-48 sys\_tags

参数	参数类型	描述
value	String	企业项目ID。
key	String	该值目前固定为“_sys_enterprise_project_id”。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/2b2a676f-0b91-4ad5-8c24-ec61be586fd1/enterprise-projects
```

## 响应示例

状态码： 200

请求成功。

```
{  
  "sys_tags": [{  
    "key": "_sys_enterprise_project_id",  
    "value": "1ce45885-4033-40d2-bdde-d4dbaceb387d"  
  }]  
}
```

## 状态码

状态码	描述
200	请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

## 4.1.9 查询集群实例信息

### 功能介绍

查询集群实例信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1.1/{project\_id}/instances/{instance\_id}

表 4-49 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	实例ID，获取方法请参见 <a href="#">获取集群列表</a> 。

### 请求参数

表 4-50 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

### 响应参数

状态码： 200

表 4-51 响应 Body 参数

参数	参数类型	描述
instance	<b>CdmQueryClusterInstanceDetail</b> object	实例信息。

表 4-52 CdmQueryClusterInstanceDetail

参数	参数类型	描述
configuration Status	String	节点配置状态： <ul style="list-style-type: none"> <li>• In-Sync: 配置已同步。</li> <li>• Applying: 配置中。</li> <li>• Sync-Failure: 配置失败。</li> </ul>
paramsGroupID	String	配置ID。
type	String	配置服务类型，这里为cdm。
role	String	实例模式，这里为Standalone。
subnetid	String	实例的子网ID。
securegroup	String	安全组ID。
vpc	String	实例的VPC ID。
azcode	String	可用区名称。
region	String	局点名称。
created	String	实例创建时间，格式为ISO8601：YYYY-MM-DDThh:mm:ssZ。
updated	String	实例更新时间，格式为ISO8601：YYYY-MM-DDThh:mm:ssZ。
name	String	实例名称。
id	String	实例ID。
flavor	<b>flavor</b> object	节点的虚拟机规格，请参见flavor参数说明。
datastore	<b>Datastore</b> object	集群信息，请参见datastore参数说明。
dbuser	String	数据库用户，这里为cdm。
payModel	Integer	付费模式： <ul style="list-style-type: none"> <li>• 0: 按需。</li> <li>• 1: 包周期。</li> </ul>

参数	参数类型	描述
publicip	String	集群绑定的公网地址。
trafficip	String	集群的内网地址。
trafficipv6	String	集群的内网IPv6地址。
cluster_id	String	集群ID。

表 4-53 flavor

参数	参数类型	描述
id	String	节点虚拟机的规格ID。
links	Array of <b>ClusterLinks</b> objects	链接信息。

表 4-54 ClusterLinks

参数	参数类型	描述
rel	String	关系。
href	String	链接地址。

表 4-55 Datastore

参数	参数类型	描述
type	String	类型，一般为cdm。
version	String	集群版本。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/instances/2c529048-ed06-4bcb-a48e-bf1800e1f496
```

## 响应示例

**状态码： 200**

请求成功。

```
{
  "instance": {
    "configurationStatus": "In-Sync",
    "paramsGroupId": "26084bb9-e74b-47d5-8be6-c0fbee9449d5",
    "type": "cdm",
```

```
"subnetid" : "9e4049b5-19a6-48fe-b5a2-2857c842fe56",
"securegroup" : "560a3642-ddb1-4e93-a4bb-e484ae975127",
"vpc" : "f35aee01-c4a3-47c1-8d92-9df430537de4",
"azcode" : "xxx-xxx-xxxa",
"region" : "xxx-xxx-xxx",
"created" : "2018-09-05T08:38:25",
"updated" : "2018-09-05T08:38:25",
"name" : "test-cdm-dn-1-1",
"id" : "2c529048-ed06-4bcb-a48e-bf1800e1f496",
"flavor" : {
  "id" : "a79fd5ae-1833-448a-88e8-3ea2b913e1f6",
  "links" : [ ]
},
"datastore" : {
  "type" : "cdm",
  "version" : "2.9.3.300"
},
"dbuser" : "cdm",
"payModel" : 0,
"publicip" : "49.xx.xx.10",
"trafficip" : "192.168.0.128",
"trafficipv6" : null,
"cluster_id" : "2d9ac57e-3ebf-4557-86d5-89ae750ff61c"
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ShowInstanceDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowInstanceDetailRequest request = new ShowInstanceDetailRequest();
        request.withInstancedId("{instance_id}");
        try {
```



```
ShowInstanceDetailResponse response = client.showInstanceDetail(request);
System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceDetailRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cdm.NewCdmClient(
    cdm.CdmClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowInstanceDetailRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowInstanceDetail(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

## 4.1.10 修改集群

### 功能介绍

修改CDM集群配置。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1.1/{project\_id}/cluster/modify/{cluster\_id}

表 4-56 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	CDM集群ID，获取方法请参见 <a href="#">查询集群列表</a> 。

### 请求参数

表 4-57 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
X-Language	是	String	请求语言。

表 4-58 请求 Body 参数

参数	是否必选	参数类型	描述
autoOff	否	Boolean	自动关机。
scheduleBootOff	否	Boolean	定时关机。
scheduleBootTime	否	String	定时开机。
scheduleOffTime	否	String	定时关机时间。
autoRemind	否	Boolean	消息通知。
phoneNum	否	String	手机号码，最多填写20个，以英文逗号分隔。
email	否	String	邮箱地址，最多填写20个，以英文逗号分隔。

## 响应参数

无

## 请求示例

修改集群配置。

```
POST /v1.1/1551c7f6c808414d8e9f3c514a170f2e/cluster/modify/bae65496-643e-47ca-84af-948672de7eeb
{
  "autoOff": false,
  "scheduleBootOff": true,
  "scheduleBootTime": "00:00:00",
  "scheduleOffTime": "10:00:00",
  "autoRemind": true,
  "phoneNum": "xxx",
  "email": "xxx@xxx.com"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

修改集群配置。

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ModifyClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ModifyClusterRequest request = new ModifyClusterRequest();
        request.withClusterId("{cluster_id}");
        CdmModifyClusterReq body = new CdmModifyClusterReq();
        body.withEmail("xxx@xxx.com");
        body.withPhoneNum("xxx");
        body.withAutoRemind(true);
        body.withScheduleOffTime("10:00:00");
        body.withScheduleBootTime("00:00:00");
        body.withScheduleBootOff(true);
        body.withAutoOff(false);
        request.withBody(body);
        try {
            ModifyClusterResponse response = client.modifyCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

修改集群配置。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ModifyClusterRequest()
        request.cluster_id = "{cluster_id}"
        request.body = CdmModifyClusterReq(
            email="xxx@xxx.com",
            phone_num="xxx",
            auto_remind=True,
            schedule_off_time="10:00:00",
            schedule_boot_time="00:00:00",
            schedule_boot_off=True,
            auto_off=False
        )
        response = client.modify_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

修改集群配置。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```

client := cdm.NewCdmClient(
    cdm.CdmClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ModifyClusterRequest{}
request.ClusterId = "{cluster_id}"
emailCdmModifyClusterReq:= "xxx@xxx.com"
phoneNumCdmModifyClusterReq:= "xxx"
autoRemindCdmModifyClusterReq:= true
scheduleOffTimeCdmModifyClusterReq:= "10:00:00"
scheduleBootTimeCdmModifyClusterReq:= "00:00:00"
scheduleBootOffCdmModifyClusterReq:= true
autoOffCdmModifyClusterReq:= false
request.Body = &model.CdmModifyClusterReq{
    Email: &emailCdmModifyClusterReq,
    PhoneNum: &phoneNumCdmModifyClusterReq,
    AutoRemind: &autoRemindCdmModifyClusterReq,
    ScheduleOffTime: &scheduleOffTimeCdmModifyClusterReq,
    ScheduleBootTime: &scheduleBootTimeCdmModifyClusterReq,
    ScheduleBootOff: &scheduleBootOffCdmModifyClusterReq,
    AutoOff: &autoOffCdmModifyClusterReq,
}
response, err := client.ModifyCluster(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
202	请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部接口异常。
503	服务不可用。

## 错误码

请参见[错误码](#)。

## 4.1.11 重启集群

### 功能介绍

重启集群接口。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1.1/{project\_id}/clusters/{cluster\_id}/action

表 4-59 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。

### 请求参数

表 4-60 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-61 请求 Body 参数

参数	是否必选	参数类型	描述
restart	是	<a href="#">restart</a> object	集群重启操作，定义哪些集群节点需要重启，请参见restart参数说明。



表 4-62 restart

参数	是否必选	参数类型	描述
restartDelayTime	否	Integer	重启时延，单位：秒。
restartMode	否	String	重启类型： <ul style="list-style-type: none"> <li>• IMMEDIATELY：立即重启。</li> <li>• FORCELY：强制重启。</li> <li>• SOFTLY：一般重启。</li> </ul> 默认值为“IMMEDIATELY”。强制重启业务进程会中断，并重启集群的虚拟机。
restartLevel	否	String	重启级别： <ul style="list-style-type: none"> <li>• SERVICE：重启服务。</li> <li>• VM：重启虚拟机。</li> </ul> 默认值为“SERVICE”。
type	是	String	集群节点类型，只支持“cdm”。
instance	否	String	预留字段，“restartLevel”为“SERVICE”时，“instance”必填，填空字符串。
group	否	String	预留字段，“restartLevel”为“SERVICE”时，“group”必填，填空字符串。

## 响应参数

状态码： 200

表 4-63 响应 Body 参数

参数	参数类型	描述
jobId	String	作业ID。

## 请求示例

重启集群。

```
POST /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/bae65496-643e-47ca-84af-948672de7eeb/action
{
  "restart": {
    "instance": "",
    "type": "cdm",
  }
}
```

```
"group" : ""  
}  
}
```

## 响应示例

状态码： 200

OK。

```
{  
  "jobId" : "ff8080815e59d92d015e5b27ccb0004d"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

重启集群。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;  
import com.huaweicloud.sdk.cdm.v1.*;  
import com.huaweicloud.sdk.cdm.v1.model.*;  
  
public class RestartClusterSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CdmClient client = CdmClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))  
            .build();  
        RestartClusterRequest request = new RestartClusterRequest();  
        request.withClusterId("{cluster_id}");  
        CdmRestartClusterReq body = new CdmRestartClusterReq();  
        CdmRestartClusterReqRestart restartbody = new CdmRestartClusterReqRestart();  
        restartbody.withType("cdm")  
            .withInstance("")  
            .withGroup("");  
        body.withRestart(restartbody);  
        request.withBody(body);  
        try {  
            RestartClusterResponse response = client.restartCluster(request);
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

重启集群。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RestartClusterRequest()
        request.cluster_id = "{cluster_id}"
        restartbody = CdmRestartClusterReqRestart(
            type="cdm",
            instance="",
            group=""
        )
        request.body = CdmRestartClusterReq(
            restart=restartbody
        )
        response = client.restart_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

重启集群。

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestartClusterRequest{}
    request.ClusterId = "{cluster_id}"
    instanceRestart := ""
    groupRestart := ""
    restartbody := &model.CdmRestartClusterReqRestart{
        Type: "cdm",
        Instance: &instanceRestart,
        Group: &groupRestart,
    }
    request.Body = &model.CdmRestartClusterReq{
        Restart: restartbody,
    }
    response, err := client.RestartCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。

状态码	描述
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.12 启动集群

#### 功能介绍

启动集群接口。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1.1/{project\_id}/clusters/{cluster\_id}/action

表 4-64 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。

#### 请求参数

表 4-65 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-66 请求 Body 参数

参数	是否必选	参数类型	描述
start	是	Object	集群启动操作，定义集群启动标识，为空对象。

## 响应参数

状态码： 200

表 4-67 响应 Body 参数

参数	参数类型	描述
jobId	Array of strings	作业ID。

## 请求示例

启动集群。

```
POST /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/bae65496-643e-47ca-84af-948672de7eeb/action
{
  "start" : { }
}
```

## 响应示例

状态码： 200

OK。

```
{
  "jobId" : [ "ff8080815e59d92d015e5b27ccb0004d" ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

启动集群。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;
```

```
public class StartClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        StartClusterRequest request = new StartClusterRequest();
        request.withClusterId("{cluster_id}");
        CdmStartClusterReq body = new CdmStartClusterReq();
        body.withStart(new Object());
        request.withBody(body);
        try {
            StartClusterResponse response = client.startCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

启动集群。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = CdmClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CdmRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = StartClusterRequest()
    request.cluster_id = "{cluster_id}"
    request.body = CdmStartClusterReq(
        start={}
    )
    response = client.start_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

启动集群。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StartClusterRequest{
        request.ClusterId = "{cluster_id}"
    }
    var startCdmStartClusterReq interface{} = make(map[string]string)
    request.Body = &model.CdmStartClusterReq{
        Start: &startCdmStartClusterReq,
    }
    response, err := client.StartCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```



## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.13 停止集群（待下线）

#### 功能介绍

停止集群接口。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1.1/{project\_id}/clusters/{cluster\_id}/action

表 4-68 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。

## 请求参数

表 4-69 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-70 请求 Body 参数

参数	是否必选	参数类型	描述
stop	是	<b>stop</b> object	集群停止操作，定义集群停止标识，请参见stop参数说明。

表 4-71 stop

参数	是否必选	参数类型	描述
stopMode	是	String	关机类型： <ul style="list-style-type: none"> <li>IMMEDIATELY：立即关机。</li> <li>GRACEFULLY：优雅关机。</li> </ul>
delayTime	否	Integer	关机时延，仅在stopMode为“GRACEFULLY”生效，单位：秒。该值为-1时，表示等待所有作业完成，并停止接受新作业。该值为大于0的任意值表示等待该时长后关机，并停止接受新作业。

## 响应参数

状态码： 200

表 4-72 响应 Body 参数

参数	参数类型	描述
jobId	Array of strings	作业ID。

## 请求示例

停止集群

```
POST /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/bae65496-643e-47ca-84af-948672de7eeb/action
{
  "stop": {
    "stopMode": "GRACEFULLY",
    "delayTime": -1
  }
}
```

## 响应示例

状态码： 200

OK。

```
{
  "jobId": [ "ff8080815e59d92d015e5b27ccb0004d" ]
}
```

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.14 创建集群

#### 功能介绍

创建集群接口。

#### 调用方法

请参见[如何调用API](#)。

## URI

POST /v1.1/{project\_id}/clusters

表 4-73 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 4-74 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
X-Language	是	String	请求语言。

表 4-75 请求 Body 参数

参数	是否必选	参数类型	描述
cluster	是	<a href="#">cluster</a> object	集群对象，请参见cluster参数说明。
auto_remind	否	Boolean	选择是否开启消息通知。开启后，支持配置20个手机号码或邮箱，作业（目前仅支持表/文件迁移的作业）失败时、EIP异常时会发送短信或邮件通知用。
phone_num	否	String	接收消息通知的手机号码。
email	否	String	接收消息通知的邮箱。

表 4-76 cluster

参数	是否必选	参数类型	描述
scheduleBoot Time	否	String	定时开机的时间，CDM集群会在每天这个时间开机。

参数	是否必选	参数类型	描述
isScheduleBootOff	否	Boolean	选择是否启用定时开关机功能。定时开关机功能和自动关机功能不可同时开启。
instances	否	Array of <b>instance</b> objects	节点列表，请参见instances参数说明。
datastore	否	<b>Datastore</b> object	集群信息，请参见datastore参数说明。
extended_properties	否	<b>ExtendedProperties</b> object	扩展属性，请参见extended_properties参数说明。
scheduleOffTime	否	String	定时关机的时间，定时关机时系统不会等待未完成的作业执行完成。
vpclId	否	String	指定虚拟私有云ID，用于集群网络配置。
name	否	String	集群名称。
sys_tags	否	Array of <b>sys_tags</b> objects	企业项目信息，请参见sys_tags参数说明。
isAutoOff	否	Boolean	选择是否启用自动关机功能，自动关机功能和定时开关机功能不可同时开启。如果选择自动关机，则当集群中无作业运行且无定时作业时，等待15分钟后集群将自动关机来帮您节约成本。

表 4-77 instance

参数	是否必选	参数类型	描述
availability_zone	是	String	集群的可用分区。可通过 <a href="https://developer.huaweicloud.com/endpoint">https://developer.huaweicloud.com/endpoint</a> 获取。
nics	是	Array of <b>nics</b> objects	网卡列表，最多两个网卡。请参见nics参数说明。

参数	是否必选	参数类型	描述
flavorRef	是	String	实例规格： <ul style="list-style-type: none"> <li>• a79fd5ae-1833-448a-88e8-3ea2b913e1f6：表示 cdm.small规格，2核CPU、4G内存的虚拟机。适合PoC验证和开发测试。</li> <li>• fb8fe666-6734-4b11-bc6c-43d11db3c745：表示 cdm.medium规格，4核CPU、8G内存的虚拟机适合单张表规模&lt;1000万条的场景。</li> <li>• 5ddb1071-c5d7-40e0-a874-8a032e81a697：表示 cdm.large规格，8核CPU、16G内存的虚拟机。适合单张表规模≥1000万条的场景。</li> <li>• 6ddb1072-c5d7-40e0-a874-8a032e81a698：表示 cdm.xlarge规格，16核CPU、32G内存的虚拟机。需要10GE高速带宽进行TB以上的数据量迁移时使用。</li> </ul>
type	是	String	节点类型，当前只有“cdm”一种类型。

表 4-78 nics

参数	是否必选	参数类型	描述
securityGroupId	是	String	安全组ID。
net-id	是	String	子网ID。

表 4-79 Datastore

参数	是否必选	参数类型	描述
type	否	String	类型，一般为cdm。
version	否	String	集群版本。

表 4-80 ExtendedProperties

参数	是否必选	参数类型	描述
workspaceId	否	String	工作空间ID。
resourceId	否	String	资源ID。
trial	否	String	是否是试用集群。

表 4-81 sys\_tags

参数	是否必选	参数类型	描述
value	是	String	企业项目ID。
key	是	String	该值目前固定为 “_sys_enterprise_project_id” 。

## 响应参数

状态码： 202

表 4-82 响应 Body 参数

参数	参数类型	描述
name	String	集群名称。
id	String	集群ID。
task	<b>Task</b> object	任务信息。
datastore	<b>Datastore</b> object	集群信息。
instances	Array of <b>ClusterInstance</b> objects	集群的节点信息。

表 4-83 Task

参数	参数类型	描述
id	String	任务id。
name	String	任务名称。

表 4-84 Datastore

参数	参数类型	描述
type	String	类型，一般为cdm。
version	String	集群版本。

表 4-85 ClusterInstance

参数	参数类型	描述
id	String	节点的虚拟机ID。
name	String	节点的虚拟机名称。
type	String	节点类型，只支持一种类型“cdm”。
shard_id	String	分片ID。

## 请求示例

创建一个1.8.10版本，集群名为cdm-ab82的CDM集群。

POST /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters

```
{
  "cluster": {
    "scheduleBootTime": "",
    "isScheduleBootOff": false,
    "instances": [ {
      "availability_zone": "xx-xxx",
      "nics": [ {
        "securityGroupId": "c37852d2-2d12-41cb-af47-65c80e995c80",
        "net-id": "2d120298-6130-44d4-a438-454912fff901"
      } ],
      "flavorRef": "5ddb1071-c5d7-40e0-a874-8a032e81a697",
      "type": "cdm"
    } ],
    "datastore": {
      "type": "cdm",
      "version": "1.8.10"
    },
    "scheduleOffTime": "",
    "vpclId": "67c06084-2212-4242-bcd4-d2144c2385a9",
    "name": "cdm-ab82",
    "sys_tags": [ {
      "value": "1ce45885-4033-40d2-bdde-d4dbaceb387d",
      "key": "_sys_enterprise_project_id"
    } ],
    "isAutoOff": false
  },
  "auto_remind": false,
  "phone_num": "",
  "email": ""
}
```

## 响应示例

状态码： 202



Accepted。

```
{
  "id": "befc862c-9286-46a0-a1d6-300d98b63aad",
  "name": "cdm-4ef213",
  "task": {
    "id": "2c9080047f1b1185017f1ef6ad0500ac",
    "name": "rdsCreateBackupJob"
  },
  "datastore": {
    "type": "cdm",
    "version": "2.9.1.100"
  },
  "instances": [ {
    "id": "b2672e7d-2faf-423f-96bb-0664cd743cfd",
    "name": "cdm-4ef213-cdm-dn-1-1",
    "type": "cdm",
    "shard_id": "dn-1"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个1.8.10版本，集群名为cdm-ab82的CDM集群。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CdmCreateClusterReq body = new CdmCreateClusterReq();
        List<SysTags> listClusterSysTags = new ArrayList<>();
```

```
listClusterSysTags.add(
    new SysTags()
        .withValue("1ce45885-4033-40d2-bdde-d4dbaceb387d")
        .withKey("_sys_enterprise_project_id")
);
Datastore datastoreCluster = new Datastore();
datastoreCluster.withType("cdm")
    .withVersion("1.8.10");
List<Nics> listInstancesNics = new ArrayList<>();
listInstancesNics.add(
    new Nics()
        .withSecurityGroupIpd("c37852d2-2d12-41cb-af47-65c80e995c80")
        .withNetId("2d120298-6130-44d4-a438-454912fff901")
);
List<Instance> listClusterInstances = new ArrayList<>();
listClusterInstances.add(
    new Instance()
        .withAvailabilityZone("xx-xxx")
        .withNics(listInstancesNics)
        .withFlavorRef("5ddb1071-c5d7-40e0-a874-8a032e81a697")
        .withType("cdm")
);
CdmCreateClusterReqCluster clusterbody = new CdmCreateClusterReqCluster();
clusterbody.withScheduleBootTime("")
    .withIsScheduleBootOff(false)
    .withInstances(listClusterInstances)
    .withDatastore(datastoreCluster)
    .withScheduleOffTime("")
    .withVpId("67c06084-2212-4242-bcd4-d2144c2385a9")
    .withName("cdm-ab82")
    .withSysTags(listClusterSysTags)
    .withIsAutoOff(false);
body.withEmail("");
body.withPhoneNum("");
body.withAutoRemind(false);
body.withCluster(clusterbody);
request.withBody(body);
try {
    CreateClusterResponse response = client.createCluster(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

创建一个1.8.10版本，集群名为cdm-ab82的CDM集群。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = CdmClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CdmRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateClusterRequest()
    listSysTagsCluster = [
        SysTags(
            value="1ce45885-4033-40d2-bdde-d4dbaceb387d",
            key="_sys_enterprise_project_id"
        )
    ]
    datastoreCluster = Datastore(
        type="cdm",
        version="1.8.10"
    )
    listNicsInstances = [
        Nics(
            security_group_id="c37852d2-2d12-41cb-af47-65c80e995c80",
            net_id="2d120298-6130-44d4-a438-454912fff901"
        )
    ]
    listInstancesCluster = [
        Instance(
            availability_zone="xx-xxx",
            nics=listNicsInstances,
            flavor_ref="5ddb1071-c5d7-40e0-a874-8a032e81a697",
            type="cdm"
        )
    ]
    clusterbody = CdmCreateClusterReqCluster(
        schedule_boot_time="",
        is_schedule_boot_off=False,
        instances=listInstancesCluster,
        datastore=datastoreCluster,
        schedule_off_time="",
        vpc_id="67c06084-2212-4242-bcd4-d2144c2385a9",
        name="cdm-ab82",
        sys_tags=listSysTagsCluster,
        is_auto_off=False
    )
    request.body = CdmCreateClusterReq(
        email="",
        phone_num="",
        auto_remind=False,
        cluster=clusterbody
    )
    response = client.create_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

创建一个1.8.10版本，集群名为cdm-ab82的CDM集群。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    var listSysTagsCluster = []model.SysTags{
        {
            Value: "1ce45885-4033-40d2-bdde-d4dbaceb387d",
            Key: "_sys_enterprise_project_id",
        },
    }
    typeDatastore := "cdm"
    versionDatastore := "1.8.10"
    datastoreCluster := &model.Datastore{
        Type: &typeDatastore,
        Version: &versionDatastore,
    }
    var listNicsInstances = []model.Nics{
        {
            SecurityGroupId: "c37852d2-2d12-41cb-af47-65c80e995c80",
            NetId: "2d120298-6130-44d4-a438-454912fff901",
        },
    }
    var listInstancesCluster = []model.Instance{
        {
            AvailabilityZone: "xx-xxx",
            Nics: listNicsInstances,
            FlavorRef: "5ddb1071-c5d7-40e0-a874-8a032e81a697",
            Type: "cdm",
        },
    }
    scheduleBootTimeCluster := ""
    isScheduleBootOffCluster := false
    scheduleOffTimeCluster := ""
    vpclIdCluster := "67c06084-2212-4242-bcd4-d2144c2385a9"
    nameCluster := "cdm-ab82"
    isAutoOffCluster := false
    clusterbody := &model.CdmCreateClusterReqCluster{
        ScheduleBootTime: &scheduleBootTimeCluster,
        IsScheduleBootOff: &isScheduleBootOffCluster,
        Instances: &listInstancesCluster,
    }
}
```

```

    Datastore: datastoreCluster,
    ScheduleOffTime: &scheduleOffTimeCluster,
    VpcId: &vpcIdCluster,
    Name: &nameCluster,
    SysTags: &listSysTagsCluster,
    IsAutoOff: &isAutoOffCluster,
  }
  emailCdmCreateClusterReq:= ""
  phoneNumCdmCreateClusterReq:= ""
  autoRemindCdmCreateClusterReq:= false
  request.Body = &model.CdmCreateClusterReq{
    Email: &emailCdmCreateClusterReq,
    PhoneNum: &phoneNumCdmCreateClusterReq,
    AutoRemind: &autoRemindCdmCreateClusterReq,
    Cluster: clusterbody,
  }
  response, err := client.CreateCluster(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
202	Accepted。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部接口异常。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.1.15 查询集群列表

#### 功能介绍

查询集群列表接口。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1.1/{project\_id}/clusters

表 4-86 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 4-87 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 200

表 4-88 响应 Body 参数

参数	参数类型	描述
clusters	Array of <a href="#">clusters</a> objects	集群列表，请参见clusters参数说明。

表 4-89 clusters

参数	参数类型	描述
customerConfig	<a href="#">CustomerConfig</a> object	用户配置信息。
datastore	<a href="#">Datastore</a> object	cdm信息。

参数	参数类型	描述
instances	Array of <b>ClusterDetail Instance</b> objects	集群的节点信息，请参见instances参数说明。
azName	String	az名称。
dbuser	String	数据库用户。
flavorName	String	规格名称。
recentEvent	Integer	事件数。
isAutoOff	Boolean	自动关机。
isScheduleBootOff	Boolean	选择是否启用定时开关机功能。定时开关机功能和自动关机功能不可同时开启。
clusterMode	String	集群模式：sharding(分片集群)。
namespace	String	命名空间。
task	<b>ClusterTask</b> object	任务信息。
publicEndpoint	String	集群绑定的EIP。
actionProgress	<b>ActionProgress</b> object	集群操作进度，任务信息，由key、value组成。key值为正在进行的任务，value值为正在进行任务的进度。示例如 "action_progress": {"SNAPSHOTTING":"16%"}。
created	String	集群创建时间，格式为ISO8601：YYYY-MM-DDThh:mm:ssZ。
bakExpectedStartTime	String	开始时间。
bakKeepDay	Integer	保留时间。
name	String	集群名称。
statusDetail	String	集群状态描述：Normal（正常）。
id	String	集群ID。
isFrozen	String	集群是否冻结：0：否，1：是。
updated	String	集群更新时间，格式为ISO8601：YYYY-MM-DDThh:mm:ssZ。

参数	参数类型	描述
status	String	集群状态： <ul style="list-style-type: none"> <li>• 100：创建中。</li> <li>• 200：正常。</li> <li>• 300：失败。</li> <li>• 303：创建失败。</li> <li>• 500：重启中。</li> <li>• 800：冻结。</li> <li>• 900：已关机。</li> <li>• 910：正在关机。</li> <li>• 920：正在开机。</li> </ul>
failedReasons	<b>FailedReasons</b> object	失败原因。集群处于正常状态时不返回。

表 4-90 CustomerConfig

参数	参数类型	描述
failureRemind	String	失败提醒。
clusterName	String	集群类型。
serviceProvider	String	服务提供。
localDisk	String	是否本地磁盘。
ssl	String	是否使用ssl。
createFrom	String	创建来源。
resourceId	String	资源ID。
flavorType	String	规格类型。
workspaceId	String	工作空间ID。
trial	String	适用。

表 4-91 Datastore

参数	参数类型	描述
type	String	类型，一般为cdm。
version	String	集群版本。



表 4-92 ClusterDetailInstance

参数	参数类型	描述
flavor	<b>flavor</b> object	节点的虚拟机规格，请参见flavor参数说明（查询集群列表时返回值为null）。
volume	<b>volume</b> object	节点的磁盘信息，请参见volume参数说明（查询集群列表时返回值为null）。
status	String	节点状态： <ul style="list-style-type: none"> <li>● 100：创建中。</li> <li>● 200：正常。</li> <li>● 300：失败。</li> <li>● 303：创建失败。</li> <li>● 400：已删除。</li> <li>● 800：冻结。</li> </ul>
actions	Array of strings	节点操作状态列表： <ul style="list-style-type: none"> <li>● REBOOTING：重启中。</li> <li>● RESTORING：恢复中。</li> <li>● REBOOT_FAILURE：重启失败。</li> </ul>
type	String	节点类型，只支持一种类型“cdm”。
id	String	节点的虚拟机ID。
name	String	节点的虚拟机名称。
isFrozen	String	节点是否冻结：0：否。1：是。
components	String	组件。
config_status	String	节点配置状态（查询集群列表时为null）： <ul style="list-style-type: none"> <li>● In-Sync：配置已同步。</li> <li>● Applying：配置中。</li> <li>● Sync-Failure：配置失败。</li> </ul>
role	String	实例角色。
group	String	分组。
links	Array of <b>ClusterLinks</b> objects	链接信息（查询集群列表时返回值为null）。
paramsGroupid	String	组件分组id。
publicip	String	公网ip。
managelp	String	管理ip。

参数	参数类型	描述
trafficIp	String	流量ip。
shard_id	String	分片id。
manage_fix_ip	String	管理修复ip。
private_ip	String	私有ip。
internal_ip	String	内部ip。
resource	Array of <b>Resource</b> objects	资源信息（查询集群列表时返回值为null）。

表 4-93 flavor

参数	参数类型	描述
id	String	节点虚拟机的规格ID。
links	Array of <b>ClusterLinks</b> objects	链接信息。

表 4-94 volume

参数	参数类型	描述
type	String	节点的磁盘类型，只支持本地磁盘。
size	Long	节点磁盘大小，单位G。

表 4-95 ClusterLinks

参数	参数类型	描述
rel	String	关系。
href	String	链接地址。

表 4-96 Resource

参数	参数类型	描述
resource_id	String	资源id。

参数	参数类型	描述
resource_type	String	资源类型：server(服务器)。

表 4-97 ClusterTask

参数	参数类型	描述
description	String	任务描述。
id	String	任务id。
name	String	任务名称。

表 4-98 ActionProgress

参数	参数类型	描述
CREATING	String	创建集群进度，例如：29%。
GROWING	String	扩容集群进度，例如：29%。
RESTORING	String	恢复集群进度，例如：29%。
SNAPSHOTTING	String	集群快照进度，例如：29%。
REPAIRING	String	修复集群进度，例如：29%。

表 4-99 FailedReasons

参数	参数类型	描述
CREATE_FAILED	<b>CREATE_FAILED</b> object	集群创建失败原因。

表 4-100 CREATE\_FAILED

参数	参数类型	描述
errorCode	String	错误码。
errorMsg	String	失败原因。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters
```

## 响应示例

状态码： 200

OK。

```
{
  "clusters": [ {
    "publicEndpoint": "49.xx.xx.10",
    "actionProgress": { },
    "created": "2018-09-05T08:38:25",
    "name": "cdm-c018",
    "statusDetail": "Normal",
    "id": "bae65496-643e-47ca-84af-948672de7eeb",
    "isFrozen": "0",
    "updated": "2018-09-05T08:38:25",
    "status": "200"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ListClustersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ListClustersRequest request = new ListClustersRequest();
        try {
            ListClustersResponse response = client.listClusters(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListClustersRequest()
        response = client.list_clusters(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cdm.NewCdmClient(
    cdm.CdmClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListClustersRequest{}
response, err := client.ListClusters(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

## 4.2 作业管理

### 4.2.1 查询作业

#### 功能介绍

查询作业接口。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/job/{job\_name}

表 4-101 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
job_name	是	String	查询多个作业用all,查询单个作业输入作业名。

表 4-102 Query 参数

参数	是否必选	参数类型	描述
filter	否	String	当“job_name”为“all”时，此参数用于模糊过滤作业。
page_no	否	Integer	指定作业页号。
page_size	否	Integer	每页作业数，值在10-100之间。
jobType	否	String	作业类型： <ul style="list-style-type: none"> <li>• jobType=NORMAL_JOB：表示查询表/文件迁移的作业。</li> <li>• jobType=BATCH_JOB：表示查询整库迁移的作业。</li> <li>• jobType=SCENARIO_JOB：表示查询场景迁移的作业。</li> <li>• 不指定该参数时，默认只查询表/文件迁移的作业。</li> </ul>

## 请求参数

表 4-103 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 200

表 4-104 响应 Body 参数

参数	参数类型	描述
total	Integer	作业数,查询单个作业时为0。
jobs	Array of <b>Job</b> objects	作业列表, 请参见jobs参数说明。
page_no	Integer	返回指定页号的作业。
page_size	Integer	每页作业数。

表 4-105 Job

参数	参数类型	描述
job_type	String	作业类型: <ul style="list-style-type: none"> <li>• NORMAL_JOB: 表/文件迁移。</li> <li>• BATCH_JOB: 整库迁移。</li> <li>• SCENARIO_JOB: 场景迁移。</li> </ul>



参数	参数类型	描述
from-connector-name	String	源端连接类型，对应的连接参数如下： <ul style="list-style-type: none"> <li>• generic-jdbc-connector: 关系数据库连接。</li> <li>• obs-connector: OBS连接。</li> <li>• hdfs-connector: HDFS连接。</li> <li>• hbase-connector: HBase连接、CloudTable连接。</li> <li>• hive-connector: Hive连接。</li> <li>• ftp-connector/sftp-connector: FTP/SFTP连接。</li> <li>• mongodb-connector: MongoDB连接。</li> <li>• redis-connector: Redis/DCS连接。</li> <li>• kafka-connector: Kafka连接。</li> <li>• dis-connector: DIS连接。</li> <li>• elasticsearch-connector: Elasticsearch/云搜索服务连接。</li> <li>• dli-connector: DLI连接。</li> <li>• http-connector: HTTP/HTTPS连接，该连接暂无连接参数。</li> <li>• dms-kafka-connector: DMSKafka连接。</li> </ul>
to-config-values	<b>ConfigValues</b> object	目的连接参数配置。根据不同目的端有不同的参数配置，具体可参考 <a href="#">目的端作业参数说明</a> 下相应的目的端参数配置。
to-link-name	String	目的端连接名称，即为通过“创建连接”接口创建的连接对应的连接名。
driver-config-values	<b>ConfigValues</b> object	作业任务参数配置。例如配置作业失败重试、抽取并发数，具体可参考 <a href="#">作业任务参数说明</a> 。
from-config-values	<b>ConfigValues</b> object	源连接参数配置。根据不同源端有不同的参数配置，具体可参考 <a href="#">源端作业参数说明</a> 下相应的源端参数配置。

参数	参数类型	描述
to-connector-name	String	<p>目的端连接类型，对应的连接参数如下：</p> <ul style="list-style-type: none"> <li>• generic-jdbc-connector: 关系数据库连接。</li> <li>• obs-connector: OBS连接。</li> <li>• hdfs-connector: HDFS连接。</li> <li>• hbase-connector: HBase连接、CloudTable连接。</li> <li>• hive-connector: Hive连接。</li> <li>• ftp-connector/sftp-connector: FTP/SFTP连接。</li> <li>• mongodb-connector: MongoDB连接。</li> <li>• redis-connector: Redis/DCS连接。</li> <li>• kafka-connector: Kafka连接。</li> <li>• dis-connector: DIS连接。</li> <li>• elasticsearch-connector: Elasticsearch/云搜索服务连接。</li> <li>• dli-connector: DLI连接。</li> <li>• http-connector: HTTP/HTTPS连接，该连接暂无连接参数。</li> <li>• dms-kafka-connector: DMSKafka连接。</li> </ul>
name	String	作业名称，长度在1到240个字符之间。
from-link-name	String	源连接名称，即为通过“创建连接”接口创建的连接对应的连接名。
creation-user	String	创建作业的用户。由系统生成，用户无需填写。
creation-date	Long	作业创建的时间，单位：毫秒。由系统生成，用户无需填写。
update-date	Long	作业最后更新的时间，单位：毫秒。由系统生成，用户无需填写。
is_incre_job	Boolean	是否是增量作业。已废弃。
flag	Integer	是否是定时作业标记，如果是定时作业则为1，否则为0。由系统根据定时任务配置生成，用户无需填写。
files_read	Integer	已读文件数。由系统生成，用户无需填写。
update-user	String	最后更新作业的用户。由系统生成，用户无需填写。
external_id	String	具体执行的作业id，如果是本地作业，则一般为"job_local1202051771_0002"形式，如果是DLI作业，则为DLI作业ID，比如"12345"。由系统生成，用户无需填写。

参数	参数类型	描述
type	String	与job_type一致，作业类型： <ul style="list-style-type: none"> <li>• NORMAL_JOB：表/文件迁移。</li> <li>• BATCH_JOB：整库迁移。</li> <li>• SCENARIO_JOB：场景迁移。</li> </ul>
execute_start_date	Long	最近一次执行任务开始时间，单位：毫秒。由系统生成，用户无需填写。
delete_rows	Integer	增量作业删除行数，已废弃。
enabled	Boolean	是否激活连接。由系统生成，用户无需填写。
bytes_written	Long	作业写入的字节。由系统生成，用户无需填写。
id	Integer	作业ID。由系统生成，用户无需填写。
is_use_sql	Boolean	用户是否使用sql。由系统根据源端抽取是否使用sql语句生成，用户无需填写。
update_rows	Integer	增量作业更新行数，已废弃。
group_name	String	组名。
bytes_read	Long	作业读取的字节。由系统生成，用户无需填写。
execute_update_date	Long	最近一次执行任务更新时间，单位：毫秒。由系统生成，用户无需填写。
write_rows	Integer	增量作业写入行数，已废弃。
rows_written	Integer	作业写入的行数。由系统生成，用户无需填写。
rows_read	Long	作业读取的行数。由系统生成，用户无需填写。
files_written	Integer	写入文件数。由系统生成，用户无需填写。
is_incrementing	Boolean	是否是增量作业，同is_incre_job，已废弃。
execute_create_date	Long	最近一次执行任务创建时间，单位：毫秒。由系统生成，用户无需填写。
status	String	作业最后的执行状态： <ul style="list-style-type: none"> <li>• BOOTING：启动中。</li> <li>• RUNNING：运行中。</li> <li>• SUCCEEDED：成功。</li> <li>• FAILED：失败。</li> <li>• NEW：未被执行。</li> </ul>

表 4-106 ConfigValues

参数	参数类型	描述
configs	Array of <b>configs</b> objects	源连接参数、目的连接参数和作业任务参数，它们的配置数据结构相同，其中“inputs”里的参数不一样，详细请参见configs数据结构说明。
extended-configs	<b>extended-configs</b> object	扩展配置，请参见extended-configs参数说明。扩展配置暂不对外开放，用户无需填写。

表 4-107 configs

参数	参数类型	描述
inputs	Array of <b>Input</b> objects	输入参数列表，列表中的每个参数为“name,value”结构，请参考inputs数据结构参数说明。在“from-config-values”数据结构中，不同的源连接类型有不同的“inputs”参数列表，请参见源端作业参数说明下的章节。在“to-config-values”数据结构中，不同的目的连接类型有不同的“inputs”参数列表，请参见目的端作业参数说明下面的子章节。在“driver-config-values”数据结构中，“inputs”具体参数请参见作业任务参数说明。
name	String	配置名称：源端作业的配置名称为“fromJobConfig”。目的端作业的配置名称为“toJobConfig”，连接的配置名称固定为“linkConfig”。
id	Integer	配置ID，由系统生成，用户无需填写。
type	String	配置类型，由系统生成，用户无需填写。值为LINK或者JOB，如果是连接管理API，则为LINK；如果是作业管理API，则为JOB。

表 4-108 Input

参数	参数类型	描述
name	String	<p>参数名:</p> <ul style="list-style-type: none"> <li>如果是连接管理API, 则以“linkConfig.”开头, 对于不同连接类型有不同的参数, 具体可参见<a href="#">连接参数说明</a>下相应连接的参数说明。</li> <li>如果是作业管理API, 对于源端连接参数, 则以“fromJobConfig.”开头, 具体可参见<a href="#">源端作业参数说明</a>下相应的源端参数说明; 对于目的端连接参数, 则以“toJobConfig.”开头, 具体可参见<a href="#">目的端作业参数说明</a>下相应的目的端参数说明; 对于作业任务参数, 请参见<a href="#">作业任务参数说明</a>下相应的任务参数说明。</li> </ul>
value	Object	参数值, 参数名对应的值, 必须填写为字符串。
type	String	值类型, 如STRING、INTEGER, 由系统设定, 用户无需填写。

表 4-109 extended-configs

参数	参数类型	描述
name	String	扩展配置名称, 暂不对外开放, 用户无需填写。
value	String	扩展配置值, 暂不对外开放, 用户无需填写。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/all?jobType=NORMAL_JOB
```

## 响应示例

状态码: 200

OK。

```
{
  "total": 1,
  "jobs": [ {
    "job_type": "NORMAL_JOB",
    "from-connector-name": "elasticsearch-connector",
    "to-config-values": {
      "configs": [ {
        "inputs": [ {
          "name": "toJobConfig.streamName",
          "value": "dis-lkGm"
        }, {
          "name": "toJobConfig.separator",
          "value": "|"
        }, {
          "name": "toJobConfig.columnList",
          "value": "1&2&3"
        }
      ]
    }
  }
]
```

```

    }],
    "name": "toJobConfig"
  }],
  },
  "to-link-name": "dis",
  "driver-config-values": {
    "configs": [ {
      "inputs": [ {
        "name": "throttlingConfig.numExtractors",
        "value": "1"
      }, {
        "name": "throttlingConfig.submitToCluster",
        "value": "false"
      }, {
        "name": "throttlingConfig.numLoaders",
        "value": "1"
      }, {
        "name": "throttlingConfig.recordDirtyData",
        "value": "false"
      }
    ]],
    "name": "throttlingConfig"
  }, {
    "inputs": [],
    "name": "jarConfig"
  }, {
    "inputs": [ {
      "name": "schedulerConfig.isSchedulerJob",
      "value": "false"
    }, {
      "name": "schedulerConfig.disposableType",
      "value": "NONE"
    }
  ]],
  "name": "schedulerConfig"
}, {
  "inputs": [],
  "name": "transformConfig"
}, {
  "inputs": [ {
    "name": "retryJobConfig.retryJobType",
    "value": "NONE"
  }
  ]],
  "name": "retryJobConfig"
}]]
},
"from-config-values": {
  "configs": [ {
    "inputs": [ {
      "name": "fromJobConfig.index",
      "value": "52est"
    }, {
      "name": "fromJobConfig.type",
      "value": "est_array"
    }, {
      "name": "fromJobConfig.columnList",
      "value": "array_f1_int:long&array_f2_text:string&array_f3_object:nested"
    }, {
      "name": "fromJobConfig.splitNestedField",
      "value": "false"
    }
  ]],
  "name": "fromJobConfig"
}]]
},
"to-connector-name": "dis-connector",
"name": "es_css",
"from-link-name": "css"
}],
"page_no": 1,
"page_size": 10
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ShowJobsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowJobsRequest request = new ShowJobsRequest();
        request.withClusterId("{cluster_id}");
        request.withJobName("{job_name}");
        try {
            ShowJobsResponse response = client.showJobs(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowJobsRequest()
        request.cluster_id = "{cluster_id}"
        request.job_name = "{job_name}"
        response = client.show_jobs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/ HuaweiCloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/ HuaweiCloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/ HuaweiCloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/ HuaweiCloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowJobsRequest{}
    request.ClusterId = "{cluster_id}"
    request.JobName = "{job_name}"
    response, err := client.ShowJobs(request)
    if err == nil {
```



```
fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。

## 错误码

请参见[错误码](#)。

## 4.2.2 删除作业

### 功能介绍

删除作业接口。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/job/{job\_name}

表 4-110 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
job_name	是	String	作业名称。

## 请求参数

表 4-111 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 500

表 4-112 响应 Body 参数

参数	参数类型	描述
errCode	String	错误码。
externalMessage	String	错误描述。

## 请求示例

```
DELETE /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/jdbc2hive
```

## 响应示例

状态码： 500

服务内部错误，具体返回错误码请参考错误码。

```
{
  "errCode": "Cdm.0100",
  "externalMessage": "Job[jdbc2hive] doesn't exist."
}
```

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。

状态码	描述
500	服务内部错误，具体返回错误码请参考错误码。

## 错误码

请参见[错误码](#)。

## 4.2.3 修改作业

### 功能介绍

修改作业接口。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/job/{job\_name}

表 4-113 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
job_name	是	String	作业名称。

### 请求参数

表 4-114 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-115 请求 Body 参数

参数	是否必选	参数类型	描述
jobs	是	Array of <b>Job</b> objects	作业列表，请参见jobs数据结构说明。

表 4-116 Job

参数	是否必选	参数类型	描述
job_type	是	String	作业类型： <ul style="list-style-type: none"> <li>• NORMAL_JOB：表/文件迁移。</li> <li>• BATCH_JOB：整库迁移。</li> <li>• SCENARIO_JOB：场景迁移。</li> </ul>

参数	是否必选	参数类型	描述
from-connector-name	是	String	<p>源端连接类型，对应的连接参数如下：</p> <ul style="list-style-type: none"> <li>• generic-jdbc-connector：关系数据库连接。</li> <li>• obs-connector：OBS连接。</li> <li>• hdfs-connector：HDFS连接。</li> <li>• hbase-connector：HBase连接、CloudTable连接。</li> <li>• hive-connector：Hive连接。</li> <li>• ftp-connector/sftp-connector：FTP/SFTP连接。</li> <li>• mongodb-connector：MongoDB连接。</li> <li>• redis-connector：Redis/DCS连接。</li> <li>• kafka-connector：Kafka连接。</li> <li>• dis-connector：DIS连接。</li> <li>• elasticsearch-connector：Elasticsearch/云搜索服务连接。</li> <li>• dli-connector：DLI连接。</li> <li>• http-connector：HTTP/HTTPS连接，该连接暂无连接参数。</li> <li>• dms-kafka-connector：DMSKafka连接。</li> </ul>
to-config-values	是	<b>ConfigValues</b> object	目的连接参数配置。根据不同目的端有不同的参数配置，具体可参考 <a href="#">目的端作业参数说明</a> 下相应的目的端参数配置。
to-link-name	是	String	目的端连接名称，即为通过“创建连接”接口创建的连接对应的连接名。
driver-config-values	是	<b>ConfigValues</b> object	作业任务参数配置。例如配置作业失败重试、抽取并发数，具体可参考 <a href="#">作业任务参数说明</a> 。

参数	是否必选	参数类型	描述
from-config-values	是	ConfigValues object	源连接参数配置。根据不同源端有不同的参数配置，具体可参考 <a href="#">源端作业参数说明</a> 下相应的源端参数配置。
to-connector-name	是	String	目的端连接类型，对应的连接参数如下： <ul style="list-style-type: none"> <li>• generic-jdbc-connector: 关系数据库连接。</li> <li>• obs-connector: OBS连接。</li> <li>• hdfs-connector: HDFS连接。</li> <li>• hbase-connector: HBase连接、CloudTable连接。</li> <li>• hive-connector: Hive连接。</li> <li>• ftp-connector/sftp-connector: FTP/SFTP连接。</li> <li>• mongodb-connector: MongoDB连接。</li> <li>• redis-connector: Redis/DCS连接。</li> <li>• kafka-connector: Kafka连接。</li> <li>• dis-connector: DIS连接。</li> <li>• elasticsearch-connector: Elasticsearch/云搜索服务连接。</li> <li>• dli-connector: DLI连接。</li> <li>• http-connector: HTTP/HTTPS连接，该连接暂无连接参数。</li> <li>• dms-kafka-connector: DMSKafka连接。</li> </ul>
name	是	String	作业名称，长度在1到240个字符之间。
from-link-name	是	String	源连接名称，即为通过“创建连接”接口创建的连接对应的连接名。
creation-user	否	String	创建作业的用户。由系统生成，用户无需填写。
creation-date	否	Long	作业创建的时间，单位：毫秒。由系统生成，用户无需填写。

参数	是否必选	参数类型	描述
update-date	否	Long	作业最后更新的时间，单位：毫秒。由系统生成，用户无需填写。
is_incre_job	否	Boolean	是否是增量作业。已废弃。
flag	否	Integer	是否是定时作业标记，如果是定时作业则为1，否则为0。由系统根据定时任务配置生成，用户无需填写。
files_read	否	Integer	已读文件数。由系统生成，用户无需填写。
update-user	否	String	最后更新作业的用户。由系统生成，用户无需填写。
external_id	否	String	具体执行的作业id，如果是本地作业，则一般为"job_local1202051771_0002"形式，如果是DLI作业，则为DLI作业ID，比如"12345"。由系统生成，用户无需填写。
type	否	String	与job_type一致，作业类型： <ul style="list-style-type: none"> <li>• NORMAL_JOB：表/文件迁移。</li> <li>• BATCH_JOB：整库迁移。</li> <li>• SCENARIO_JOB：场景迁移。</li> </ul>
execute_start_date	否	Long	最近一次执行任务开始时间，单位：毫秒。由系统生成，用户无需填写。
delete_rows	否	Integer	增量作业删除行数，已废弃。
enabled	否	Boolean	是否激活连接。由系统生成，用户无需填写。
bytes_written	否	Long	作业写入的字节。由系统生成，用户无需填写。
id	否	Integer	作业ID。由系统生成，用户无需填写。
is_use_sql	否	Boolean	用户是否使用sql。由系统根据源端抽取是否使用sql语句生成，用户无需填写。
update_rows	否	Integer	增量作业更新行数，已废弃。
group_name	否	String	组名。

参数	是否必选	参数类型	描述
bytes_read	否	Long	作业读取的字节。由系统生成，用户无需填写。
execute_update_date	否	Long	最近一次执行任务更新时间，单位：毫秒。由系统生成，用户无需填写。
write_rows	否	Integer	增量作业写入行数，已废弃。
rows_written	否	Integer	作业写入的行数。由系统生成，用户无需填写。
rows_read	否	Long	作业读取的行数。由系统生成，用户无需填写。
files_written	否	Integer	写入文件数。由系统生成，用户无需填写。
is_incrementing	否	Boolean	是否是增量作业，同 is_incre_job，已废弃。
execute_create_date	否	Long	最近一次执行任务创建时间，单位：毫秒。由系统生成，用户无需填写。
status	否	String	作业最后的执行状态： <ul style="list-style-type: none"> <li>• BOOTING：启动中。</li> <li>• RUNNING：运行中。</li> <li>• SUCCEEDED：成功。</li> <li>• FAILED：失败。</li> <li>• NEW：未被执行。</li> </ul>

表 4-117 ConfigValues

参数	是否必选	参数类型	描述
configs	是	Array of <b>configs</b> objects	源连接参数、目的连接参数和作业任务参数，它们的配置数据结构相同，其中“inputs”里的参数不一样，详细请参见configs数据结构说明。
extended-configs	否	<b>extended-configs</b> object	扩展配置，请参见extended-configs参数说明。扩展配置暂不对外开放，用户无需填写。



表 4-118 configs

参数	是否必选	参数类型	描述
inputs	是	Array of <b>Input</b> objects	输入参数列表，列表中的每个参数为“name,value”结构，请参考inputs数据结构参数说明。在“from-config-values”数据结构中，不同的源连接类型有不同的“inputs”参数列表，请参见源端作业参数说明下的章节。在“to-config-values”数据结构中，不同的目的连接类型有不同的“inputs”参数列表，请参见目的端作业参数说明下面的子章节。在“driver-config-values”数据结构中，“inputs”具体参数请参见作业任务参数说明。
name	是	String	配置名称：源端作业的配置名称为“fromJobConfig”。目的端作业的配置名称为“toJobConfig”，连接的配置名称固定为“linkConfig”。
id	否	Integer	配置ID，由系统生成，用户无需填写。
type	否	String	配置类型，由系统生成，用户无需填写。值为LINK或者JOB，如果是连接管理API，则为LINK；如果是作业管理API，则为JOB。

表 4-119 Input

参数	是否必选	参数类型	描述
name	是	String	<p>参数名:</p> <ul style="list-style-type: none"> <li>如果是连接管理API, 则以“linkConfig.”开头, 对于不同连接类型有不同的参数, 具体可参见<a href="#">连接参数说明</a>下相应连接的参数说明。</li> <li>如果是作业管理API, 对于源端连接参数, 则以“fromJobConfig.”开头, 具体可参见<a href="#">源端作业参数说明</a>下相应的源端参数说明; 对于目的端连接参数, 则以“toJobConfig.”开头, 具体可参见<a href="#">目的端作业参数说明</a>下相应的目的端参数说明; 对于作业任务参数, 请参见<a href="#">作业任务参数说明</a>下相应的任务参数说明。</li> </ul>
value	是	Object	参数值, 参数名对应的值, 必须填写为字符串。
type	否	String	值类型, 如STRING、INTEGER, 由系统设定, 用户无需填写。

表 4-120 extended-configs

参数	是否必选	参数类型	描述
name	否	String	扩展配置名称, 暂不对外开放, 用户无需填写。
value	否	String	扩展配置值, 暂不对外开放, 用户无需填写。

## 响应参数

状态码: 200

表 4-121 响应 Body 参数

参数	参数类型	描述
validation-result	Array of <b>JobValidationResult</b> objects	校验结果：如果修改失败，返回失败原因。如果修改成功，返回空列表。

表 4-122 JobValidationResult

参数	参数类型	描述
message	String	错误描述。
status	String	错误级别，如：ERROR、WARNING。

状态码： 400

表 4-123 响应 Body 参数

参数	参数类型	描述
code	String	返回编码。
errCode	String	错误码。
message	String	报错信息。
externalMessage	String	附加信息。

## 请求示例

修改一个源端为Elasticsearch数据连接，目的端为DIS数据连接，作业名为es\_css的表数据迁移作业。

```
PUT /v1.1/1551c7f6c808414d8e9f3c514a170f2e/cluster/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/es_css
```

```
{
  "jobs": [ {
    "job_type": "NORMAL_JOB",
    "from-connector-name": "elasticsearch-connector",
    "to-config-values": {
      "configs": [ {
        "inputs": [ {
          "name": "toJobConfig.streamName",
          "value": "dis-lkGm"
        } ],
        "name": "toJobConfig.separator",
        "value": "|"
      }, {
        "name": "toJobConfig.columnList",
```

```

        "value" : "1&2&3"
    },
    "name" : "toJobConfig"
  }
],
"to-link-name" : "dis",
"driver-config-values" : {
  "configs" : [ {
    "inputs" : [ {
      "name" : "throttlingConfig.numExtractors",
      "value" : "1"
    }, {
      "name" : "throttlingConfig.submitToCluster",
      "value" : "false"
    }, {
      "name" : "throttlingConfig.numLoaders",
      "value" : "1"
    }, {
      "name" : "throttlingConfig.recordDirtyData",
      "value" : "false"
    }
  ],
  "name" : "throttlingConfig"
}, {
  "inputs" : [ ],
  "name" : "jarConfig"
}, {
  "inputs" : [ {
    "name" : "schedulerConfig.isSchedulerJob",
    "value" : "false"
  }, {
    "name" : "schedulerConfig.disposableType",
    "value" : "NONE"
  }
  ],
  "name" : "schedulerConfig"
}, {
  "inputs" : [ ],
  "name" : "transformConfig"
}, {
  "inputs" : [ {
    "name" : "retryJobConfig.retryJobType",
    "value" : "NONE"
  }
  ],
  "name" : "retryJobConfig"
}
]
},
"from-config-values" : {
  "configs" : [ {
    "inputs" : [ {
      "name" : "fromJobConfig.index",
      "value" : "52est"
    }, {
      "name" : "fromJobConfig.type",
      "value" : "est_array"
    }, {
      "name" : "fromJobConfig.columnList",
      "value" : "array_f1_int:long&array_f2_text:string&array_f3_object:nested"
    }, {
      "name" : "fromJobConfig.splitNestedField",
      "value" : "false"
    }
  ],
  "name" : "fromJobConfig"
}
]
},
"to-connector-name" : "dis-connector",
"name" : "es_css",
"from-link-name" : "css"
}
}
}

```

## 响应示例

**状态码： 200**

OK。

```
{
  "validation-result" : [{}, {}, {}]
}
```

**状态码： 400**

报错错误码。

```
{
  "code" : "Cdm.0095",
  "errCode" : "Cdm.00095",
  "message" : "A job with the name obs-obs does not exist.",
  "externalMessage" : "A job with the name obs-obs does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

修改一个源端为Elasticsearch数据连接，目的端为DIS数据连接，作业名为es\_css的表数据迁移作业。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateJobSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateJobRequest request = new UpdateJobRequest();
```

```
request.withClusterId("{cluster_id}");
request.withJobName("{job_name}");
CdmUpdateJobJsonReq body = new CdmUpdateJobJsonReq();
List<Input> listConfigsInputs = new ArrayList<>();
listConfigsInputs.add(
    new Input()
        .withName("fromJobConfig.index")
        .withValue("52est")
);
listConfigsInputs.add(
    new Input()
        .withName("fromJobConfig.type")
        .withValue("est_array")
);
listConfigsInputs.add(
    new Input()
        .withName("fromJobConfig.columnList")
        .withValue("array_f1_int:long&array_f2_text:string&array_f3_object:nested")
);
listConfigsInputs.add(
    new Input()
        .withName("fromJobConfig.splitNestedField")
        .withValue("false")
);
List<Configs> listFromConfigValuesConfigs = new ArrayList<>();
listFromConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs)
        .withName("fromJobConfig")
);
ConfigValues fromconfigvaluesJobs = new ConfigValues();
fromconfigvaluesJobs.withConfigs(listFromConfigValuesConfigs);
List<Input> listConfigsInputs1 = new ArrayList<>();
listConfigsInputs1.add(
    new Input()
        .withName("retryJobConfig.retryJobType")
        .withValue("NONE")
);
List<Input> listConfigsInputs2 = new ArrayList<>();
listConfigsInputs2.add(
    new Input()
        .withName("schedulerConfig.isSchedulerJob")
        .withValue("false")
);
listConfigsInputs2.add(
    new Input()
        .withName("schedulerConfig.disposableType")
        .withValue("NONE")
);
List<Input> listConfigsInputs3 = new ArrayList<>();
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.numExtractors")
        .withValue("1")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.submitToCluster")
        .withValue("false")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.numLoaders")
        .withValue("1")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.recordDirtyData")
        .withValue("false")
);
```

```
);
List<Configs> listDriverConfigValuesConfigs = new ArrayList<>();
listDriverConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs1)
        .withName("retryJobConfig")
);
ConfigValues driverconfigvaluesJobs = new ConfigValues();
driverconfigvaluesJobs.withConfigs(listDriverConfigValuesConfigs);
List<Input> listConfigsInputs4 = new ArrayList<>();
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.streamName")
        .withValue("dis-lkGm")
);
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.separator")
        .withValue("|")
);
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.columnList")
        .withValue("1&2&3")
);
List<Configs> listToConfigValuesConfigs = new ArrayList<>();
listToConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs4)
        .withName("toJobConfig")
);
ConfigValues toconfigvaluesJobs = new ConfigValues();
toconfigvaluesJobs.withConfigs(listToConfigValuesConfigs);
List<Job> listbodyJobs = new ArrayList<>();
listbodyJobs.add(
    new Job()
        .withJobType(Job.JobTypeEnum.fromValue("NORMAL_JOB"))
        .withFromConnectorName("elasticsearch-connector")
        .withToConfigValues(toconfigvaluesJobs)
        .withToLinkName("dis")
        .withDriverConfigValues(driverconfigvaluesJobs)
        .withFromConfigValues(fromconfigvaluesJobs)
        .withToConnectorName("dis-connector")
        .withName("es_css")
        .withFromLinkName("css")
);
body.withJobs(listbodyJobs);
request.withBody(body);
try {
    UpdateJobResponse response = client.updateJob(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

修改一个源端为Elasticsearch数据连接，目的端为DIS数据连接，作业名为es\_css的表数据迁移作业。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateJobRequest()
        request.cluster_id = "{cluster_id}"
        request.job_name = "{job_name}"
        listInputsConfigs = [
            Input(
                name="fromJobConfig.index",
                value="52est"
            ),
            Input(
                name="fromJobConfig.type",
                value="est_array"
            ),
            Input(
                name="fromJobConfig.columnList",
                value="array_f1_int:long&array_f2_text:string&array_f3_object:nested"
            ),
            Input(
                name="fromJobConfig.splitNestedField",
                value="false"
            )
        ]
        listConfigsFromconfigvalues = [
            Configs(
                inputs=listInputsConfigs,
                name="fromJobConfig"
            )
        ]
        fromconfigvaluesJobs = ConfigValues(
            configs=listConfigsFromconfigvalues
        )
        listInputsConfigs1 = [
            Input(
                name="retryJobConfig.retryJobType",
                value="NONE"
            )
        ]
        listInputsConfigs2 = [
            Input(
```



```

        name="schedulerConfig.isSchedulerJob",
        value="false"
    ),
    Input(
        name="schedulerConfig.disposableType",
        value="NONE"
    )
]
listInputsConfigs3 = [
    Input(
        name="throttlingConfig.numExtractors",
        value="1"
    ),
    Input(
        name="throttlingConfig.submitToCluster",
        value="false"
    ),
    Input(
        name="throttlingConfig.numLoaders",
        value="1"
    ),
    Input(
        name="throttlingConfig.recordDirtyData",
        value="false"
    )
]
listConfigsDriverconfigvalues = [
    Configs(
        inputs=listInputsConfigs1,
        name="retryJobConfig"
    )
]
driverconfigvaluesJobs = ConfigValues(
    configs=listConfigsDriverconfigvalues
)
listInputsConfigs4 = [
    Input(
        name="toJobConfig.streamName",
        value="dis-lkGm"
    ),
    Input(
        name="toJobConfig.separator",
        value="|"
    ),
    Input(
        name="toJobConfig.columnList",
        value="1&2&3"
    )
]
listConfigsToconfigvalues = [
    Configs(
        inputs=listInputsConfigs4,
        name="toJobConfig"
    )
]
toconfigvaluesJobs = ConfigValues(
    configs=listConfigsToconfigvalues
)
listJobsbody = [
    Job(
        job_type="NORMAL_JOB",
        from_connector_name="elasticsearch-connector",
        to_config_values=toconfigvaluesJobs,
        to_link_name="dis",
        driver_config_values=driverconfigvaluesJobs,
        from_config_values=fromconfigvaluesJobs,
        to_connector_name="dis-connector",
        name="es_css",
        from_link_name="css"
    )
]

```

```
)
]
request.body = CdmUpdateJobJsonReq(
    jobs=listJobsbody
)
response = client.update_job(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

修改一个源端为Elasticsearch数据连接，目的端为DIS数据连接，作业名为es\_css的表数据迁移作业。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateJobRequest{}
    request.ClusterId = "{cluster_id}"
    request.JobName = "{job_name}"
    valueInputs := "52est"
    var valueInputsInterface interface{} = valueInputs
    valueInputs1 := "est_array"
    var valueInputsInterface1 interface{} = valueInputs1
    valueInputs2 := "array_f1_int:long&array_f2_text:string&array_f3_object:nested"
    var valueInputsInterface2 interface{} = valueInputs2
    valueInputs3 := "false"
    var valueInputsInterface3 interface{} = valueInputs3
    var listInputsConfigs = []model.Input{
        {
            Name: "fromJobConfig.index",
            Value: &valueInputsInterface,
        },
        {
            Name: "fromJobConfig.type",
            Value: &valueInputsInterface1,
        },
    }
}
```

```

    },
    {
      Name: "fromJobConfig.columnList",
      Value: &valueInputsInterface2,
    },
    {
      Name: "fromJobConfig.splitNestedField",
      Value: &valueInputsInterface3,
    },
  },
}
var listConfigsFromConfigValues = []model.Configs{
  {
    Inputs: listInputsConfigs,
    Name: "fromJobConfig",
  },
}
fromconfigvaluesJobs := &model.ConfigValues{
  Configs: listConfigsFromConfigValues,
}
valueInputs4:= "NONE"
var valueInputsInterface4 interface{} = valueInputs4
var listInputsConfigs1 = []model.Input{
  {
    Name: "retryJobConfig.retryJobType",
    Value: &valueInputsInterface4,
  },
}
valueInputs5:= "false"
var valueInputsInterface5 interface{} = valueInputs5
valueInputs6:= "NONE"
var valueInputsInterface6 interface{} = valueInputs6
var listInputsConfigs2 = []model.Input{
  {
    Name: "schedulerConfig.isSchedulerJob",
    Value: &valueInputsInterface5,
  },
  {
    Name: "schedulerConfig.disposableType",
    Value: &valueInputsInterface6,
  },
}
valueInputs7:= "1"
var valueInputsInterface7 interface{} = valueInputs7
valueInputs8:= "false"
var valueInputsInterface8 interface{} = valueInputs8
valueInputs9:= "1"
var valueInputsInterface9 interface{} = valueInputs9
valueInputs10:= "false"
var valueInputsInterface10 interface{} = valueInputs10
var listInputsConfigs3 = []model.Input{
  {
    Name: "throttlingConfig.numExtractors",
    Value: &valueInputsInterface7,
  },
  {
    Name: "throttlingConfig.submitToCluster",
    Value: &valueInputsInterface8,
  },
  {
    Name: "throttlingConfig.numLoaders",
    Value: &valueInputsInterface9,
  },
  {
    Name: "throttlingConfig.recordDirtyData",
    Value: &valueInputsInterface10,
  },
}
var listConfigsDriverConfigValues = []model.Configs{
  {

```

```

        Inputs: listInputsConfigs1,
        Name: "retryJobConfig",
    },
}
driverconfigvaluesJobs := &model.ConfigValues{
    Configs: listConfigsDriverConfigValues,
}
valueInputs11:= "dis-lkGm"
var valueInputsInterface11 interface{} = valueInputs11
valueInputs12:= "|"
var valueInputsInterface12 interface{} = valueInputs12
valueInputs13:= "1&2&3"
var valueInputsInterface13 interface{} = valueInputs13
var listInputsConfigs4 = []model.Input{
    {
        Name: "toJobConfig.streamName",
        Value: &valueInputsInterface11,
    },
    {
        Name: "toJobConfig.separator",
        Value: &valueInputsInterface12,
    },
    {
        Name: "toJobConfig.columnList",
        Value: &valueInputsInterface13,
    },
}
var listConfigsToConfigValues = []model.Configs{
    {
        Inputs: listInputsConfigs4,
        Name: "toJobConfig",
    },
}
toconfigvaluesJobs := &model.ConfigValues{
    Configs: listConfigsToConfigValues,
}
var listJobsbody = []model.Job{
    {
        JobType: model.GetJobJobTypeEnum().NORMAL_JOB,
        FromConnectorName: "elasticsearch-connector",
        ToConfigValues: toconfigvaluesJobs,
        ToLinkName: "dis",
        DriverConfigValues: driverconfigvaluesJobs,
        FromConfigValues: fromconfigvaluesJobs,
        ToConnectorName: "dis-connector",
        Name: "es_css",
        FromLinkName: "css",
    },
}
request.Body = &model.CdmUpdateJobJsonReq{
    Jobs: listJobsbody,
}
response, err := client.UpdateJob(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	报错错误码。

## 错误码

请参见[错误码](#)。

## 4.2.4 随机集群创建作业并执行

### 功能介绍

随机集群创建作业并执行接口。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1.1/{project\_id}/clusters/job

表 4-124 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 4-125 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
X-Language	是	String	请求语言。

表 4-126 请求 Body 参数

参数	是否必选	参数类型	描述
jobs	是	Array of <b>Job</b> objects	作业列表，请参见jobs数据结构说明。
clusters	是	Array of strings	CDM集群ID列表，系统会从里面随机选择一个开机状态的集群，在该集群中创建作业并执行作业。

表 4-127 Job

参数	是否必选	参数类型	描述
job_type	是	String	作业类型： <ul style="list-style-type: none"> <li>• NORMAL_JOB：表/文件迁移。</li> <li>• BATCH_JOB：整库迁移。</li> <li>• SCENARIO_JOB：场景迁移。</li> </ul>

参数	是否必选	参数类型	描述
from-connector-name	是	String	<p>源端连接类型，对应的连接参数如下：</p> <ul style="list-style-type: none"> <li>• generic-jdbc-connector：关系数据库连接。</li> <li>• obs-connector：OBS连接。</li> <li>• hdfs-connector：HDFS连接。</li> <li>• hbase-connector：HBase连接、CloudTable连接。</li> <li>• hive-connector：Hive连接。</li> <li>• ftp-connector/sftp-connector：FTP/SFTP连接。</li> <li>• mongodb-connector：MongoDB连接。</li> <li>• redis-connector：Redis/DCS连接。</li> <li>• kafka-connector：Kafka连接。</li> <li>• dis-connector：DIS连接。</li> <li>• elasticsearch-connector：Elasticsearch/云搜索服务连接。</li> <li>• dli-connector：DLI连接。</li> <li>• http-connector：HTTP/HTTPS连接，该连接暂无连接参数。</li> <li>• dms-kafka-connector：DMSKafka连接。</li> </ul>
to-config-values	是	<b>ConfigValues</b> object	目的连接参数配置。根据不同目的端有不同的参数配置，具体可参考 <a href="#">目的端作业参数说明</a> 下相应的目的端参数配置。
to-link-name	是	String	目的端连接名称，即为通过“创建连接”接口创建的连接对应的连接名。
driver-config-values	是	<b>ConfigValues</b> object	作业任务参数配置。例如配置作业失败重试、抽取并发数，具体可参考 <a href="#">作业任务参数说明</a> 。

参数	是否必选	参数类型	描述
from-config-values	是	ConfigValues object	源连接参数配置。根据不同源端有不同的参数配置，具体可参考 <a href="#">源端作业参数说明</a> 下相应的源端参数配置。
to-connector-name	是	String	目的端连接类型，对应的连接参数如下： <ul style="list-style-type: none"> <li>• generic-jdbc-connector: 关系数据库连接。</li> <li>• obs-connector: OBS连接。</li> <li>• hdfs-connector: HDFS连接。</li> <li>• hbase-connector: HBase连接、CloudTable连接。</li> <li>• hive-connector: Hive连接。</li> <li>• ftp-connector/sftp-connector: FTP/SFTP连接。</li> <li>• mongodb-connector: MongoDB连接。</li> <li>• redis-connector: Redis/DCS连接。</li> <li>• kafka-connector: Kafka连接。</li> <li>• dis-connector: DIS连接。</li> <li>• elasticsearch-connector: Elasticsearch/云搜索服务连接。</li> <li>• dli-connector: DLI连接。</li> <li>• http-connector: HTTP/HTTPS连接，该连接暂无连接参数。</li> <li>• dms-kafka-connector: DMSKafka连接。</li> </ul>
name	是	String	作业名称，长度在1到240个字符之间。
from-link-name	是	String	源连接名称，即为通过“创建连接”接口创建的连接对应的连接名。
creation-user	否	String	创建作业的用户。由系统生成，用户无需填写。
creation-date	否	Long	作业创建的时间，单位：毫秒。由系统生成，用户无需填写。



参数	是否必选	参数类型	描述
update-date	否	Long	作业最后更新的时间，单位：毫秒。由系统生成，用户无需填写。
is_incre_job	否	Boolean	是否是增量作业。已废弃。
flag	否	Integer	是否是定时作业标记，如果是定时作业则为1，否则为0。由系统根据定时任务配置生成，用户无需填写。
files_read	否	Integer	已读文件数。由系统生成，用户无需填写。
update-user	否	String	最后更新作业的用户。由系统生成，用户无需填写。
external_id	否	String	具体执行的作业id，如果是本地作业，则一般为"job_local1202051771_0002"形式，如果是DLI作业，则为DLI作业ID，比如"12345"。由系统生成，用户无需填写。
type	否	String	与job_type一致，作业类型： <ul style="list-style-type: none"> <li>• NORMAL_JOB：表/文件迁移。</li> <li>• BATCH_JOB：整库迁移。</li> <li>• SCENARIO_JOB：场景迁移。</li> </ul>
execute_start_date	否	Long	最近一次执行任务开始时间，单位：毫秒。由系统生成，用户无需填写。
delete_rows	否	Integer	增量作业删除行数，已废弃。
enabled	否	Boolean	是否激活连接。由系统生成，用户无需填写。
bytes_written	否	Long	作业写入的字节。由系统生成，用户无需填写。
id	否	Integer	作业ID。由系统生成，用户无需填写。
is_use_sql	否	Boolean	用户是否使用sql。由系统根据源端抽取是否使用sql语句生成，用户无需填写。
update_rows	否	Integer	增量作业更新行数，已废弃。
group_name	否	String	组名。

参数	是否必选	参数类型	描述
bytes_read	否	Long	作业读取的字节。由系统生成，用户无需填写。
execute_update_date	否	Long	最近一次执行任务更新时间，单位：毫秒。由系统生成，用户无需填写。
write_rows	否	Integer	增量作业写入行数，已废弃。
rows_written	否	Integer	作业写入的行数。由系统生成，用户无需填写。
rows_read	否	Long	作业读取的行数。由系统生成，用户无需填写。
files_written	否	Integer	写入文件数。由系统生成，用户无需填写。
is_incrementing	否	Boolean	是否是增量作业，同 is_incre_job，已废弃。
execute_create_date	否	Long	最近一次执行任务创建时间，单位：毫秒。由系统生成，用户无需填写。
status	否	String	作业最后的执行状态： <ul style="list-style-type: none"> <li>• BOOTING：启动中。</li> <li>• RUNNING：运行中。</li> <li>• SUCCEEDED：成功。</li> <li>• FAILED：失败。</li> <li>• NEW：未被执行。</li> </ul>

表 4-128 ConfigValues

参数	是否必选	参数类型	描述
configs	是	Array of <b>configs</b> objects	源连接参数、目的连接参数和作业任务参数，它们的配置数据结构相同，其中“inputs”里的参数不一样，详细请参见configs数据结构说明。
extended-configs	否	<b>extended-configs</b> object	扩展配置，请参见extended-configs参数说明。扩展配置暂不对外开放，用户无需填写。

表 4-129 configs

参数	是否必选	参数类型	描述
inputs	是	Array of <b>Input</b> objects	输入参数列表，列表中的每个参数为“name,value”结构，请参考inputs数据结构参数说明。在“from-config-values”数据结构中，不同的源连接类型有不同的“inputs”参数列表，请参见源端作业参数说明下的章节。在“to-config-values”数据结构中，不同的目的连接类型有不同的“inputs”参数列表，请参见目的端作业参数说明下面的子章节。在“driver-config-values”数据结构中，“inputs”具体参数请参见作业任务参数说明。
name	是	String	配置名称：源端作业的配置名称为“fromJobConfig”。目的端作业的配置名称为“toJobConfig”，连接的配置名称固定为“linkConfig”。
id	否	Integer	配置ID，由系统生成，用户无需填写。
type	否	String	配置类型，由系统生成，用户无需填写。值为LINK或者JOB，如果是连接管理API，则为LINK；如果是作业管理API，则为JOB。

表 4-130 Input

参数	是否必选	参数类型	描述
name	是	String	<p>参数名:</p> <ul style="list-style-type: none"> <li>如果是连接管理API, 则以“linkConfig.”开头, 对于不同连接类型有不同的参数, 具体可参见<a href="#">连接参数说明</a>下相应连接的参数说明。</li> <li>如果是作业管理API, 对于源端连接参数, 则以“fromJobConfig.”开头, 具体可参见<a href="#">源端作业参数说明</a>下相应的源端参数说明; 对于目的端连接参数, 则以“toJobConfig.”开头, 具体可参见<a href="#">目的端作业参数说明</a>下相应的目的端参数说明; 对于作业任务参数, 请参见<a href="#">作业任务参数说明</a>下相应的任务参数说明。</li> </ul>
value	是	Object	参数值, 参数名对应的值, 必须填写为字符串。
type	否	String	值类型, 如STRING、INTEGER, 由系统设定, 用户无需填写。

表 4-131 extended-configs

参数	是否必选	参数类型	描述
name	否	String	扩展配置名称, 暂不对外开放, 用户无需填写。
value	否	String	扩展配置值, 暂不对外开放, 用户无需填写。

## 响应参数

状态码: 200

表 4-132 响应 Body 参数

参数	参数类型	描述
submissions	Array of <a href="#">StartJobSubmission</a> objects	作业运行信息，请参见 <a href="#">submission</a> 参数说明。

表 4-133 StartJobSubmission

参数	参数类型	描述
isIncrementing	Boolean	作业是否为增量迁移。
delete_rows	Integer	删除数据行数。
update_rows	Integer	更新数据行数。
write_rows	Integer	写入数据行数。
submission-id	Integer	作业提交id。
job-name	String	作业名称。
creation-user	String	创建用户。
creation-date	Long	创建时间，单位：毫秒。
execute-date	Long	执行时间。
progress	Float	作业进度，失败时为“-1”，其它情况为0~100。
status	String	作业状态： <ul style="list-style-type: none"> <li>● BOOTING：启动中。</li> <li>● FAILURE_ON_SUBMIT：提交失败。</li> <li>● RUNNING：运行中。</li> <li>● SUCCEEDED：成功。</li> <li>● FAILED：失败。</li> <li>● UNKNOWN：未知。</li> <li>● NEVER_EXECUTED：未被执行。</li> </ul>
isStoppingIncrement	String	是否停止增量迁移。
is-execute-auto	Boolean	是否定时执行作业。
last-update-date	Long	作业最后更新时间。

参数	参数类型	描述
last-udpate-user	String	最后更新作业状态的用户。
isDeleteJob	Boolean	作业执行完成后是否删除。

## 请求示例

在CDM集群ID列表中随机选择一个集群，创建一个源端为elasticsearch，目的端为DIS，作业名为es\_css的表迁移作业。

POST /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/job

```
{
  "jobs": [ {
    "job_type": "NORMAL_JOB",
    "from-connector-name": "elasticsearch-connector",
    "to-config-values": {
      "configs": [ {
        "inputs": [ {
          "name": "toJobConfig.streamName",
          "value": "dis-lkGm"
        }, {
          "name": "toJobConfig.separator",
          "value": "|"
        }, {
          "name": "toJobConfig.columnList",
          "value": "1&2&3"
        }
      ],
      "name": "toJobConfig"
    }
  ],
  "to-link-name": "dis",
  "driver-config-values": {
    "configs": [ {
      "inputs": [ {
        "name": "throttlingConfig.numExtractors",
        "value": "1"
      }, {
        "name": "throttlingConfig.submitToCluster",
        "value": "false"
      }, {
        "name": "throttlingConfig.numLoaders",
        "value": "1"
      }, {
        "name": "throttlingConfig.recordDirtyData",
        "value": "false"
      }
    ],
    "name": "throttlingConfig"
  }, {
    "inputs": [ ],
    "name": "jarConfig"
  }, {
    "inputs": [ {
      "name": "schedulerConfig.isSchedulerJob",
      "value": "false"
    }, {
      "name": "schedulerConfig.disposableType",
      "value": "NONE"
    }
  ],
  "name": "schedulerConfig"
  }, {
    "inputs": [ ],
    "name": "transformConfig"
  }
  ]
}
```

```

    }, {
      "inputs": [ {
        "name": "retryJobConfig.retryJobType",
        "value": "NONE"
      } ],
      "name": "retryJobConfig"
    } ]
  },
  "from-config-values": {
    "configs": [ {
      "inputs": [ {
        "name": "fromJobConfig.index",
        "value": "52est"
      } ],
      {
        "name": "fromJobConfig.type",
        "value": "est_array"
      },
      {
        "name": "fromJobConfig.columnList",
        "value": "array_f1_int:long&array_f2_text:string&array_f3_object:nested"
      },
      {
        "name": "fromJobConfig.splitNestedField",
        "value": "false"
      }
    ] ],
    "name": "fromJobConfig"
  } ]
},
"to-connector-name": "dis-connector",
"name": "es_css",
"from-link-name": "css"
}],
"clusters": [ "b0791496-e111-4e75-b7ca-9277aeab9297", "c2db1191-eb6c-464a-a0d3-b434e6c6df26",
"c2db1191-eb6c-464a-a0d3-b434e6c6df26" ]
}

```

## 响应示例

**状态码： 200**

OK。

```

{
  "submissions": [ {
    "isIncrementing": false,
    "job-name": "obs2obs-03",
    "submission-id": 13,
    "isStoppingIncrement": "",
    "last-update-date": 1635909057030,
    "is-execute-auto": false,
    "delete_rows": 0,
    "write_rows": 0,
    "isDeleteJob": false,
    "creation-user": "cdmUser",
    "progress": 0,
    "creation-date": 1635909057030,
    "update_rows": 0,
    "status": "PENDING",
    "execute-date": 1635909057030
  } ]
}

```

## SDK 代码示例

SDK代码示例如下。

## Java

在CDM集群ID列表中随机选择一个集群，创建一个源端为elasticsearch，目的端为DIS，作业名为es\_css的表迁移作业。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateAndStartRandomClusterJobSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateAndStartRandomClusterJobRequest request = new CreateAndStartRandomClusterJobRequest();
        CdmRandomCreateAndStartJobJsonReq body = new CdmRandomCreateAndStartJobJsonReq();
        List<String> listbodyClusters = new ArrayList<>();
        listbodyClusters.add("b0791496-e111-4e75-b7ca-9277aeab9297");
        listbodyClusters.add("c2db1191-eb6c-464a-a0d3-b434e6c6df26");
        listbodyClusters.add("c2db1191-eb6c-464a-a0d3-b434e6c6df26");
        List<Input> listConfigsInputs = new ArrayList<>();
        listConfigsInputs.add(
            new Input()
                .withName("fromJobConfig.index")
                .withValue("52est")
        );
        listConfigsInputs.add(
            new Input()
                .withName("fromJobConfig.type")
                .withValue("est_array")
        );
        listConfigsInputs.add(
            new Input()
                .withName("fromJobConfig.columnList")
                .withValue("array_f1_int:long&array_f2_text:string&array_f3_object:nested")
        );
        listConfigsInputs.add(
            new Input()
                .withName("fromJobConfig.splitNestedField")
                .withValue("false")
        );
        List<Configs> listFromConfigValuesConfigs = new ArrayList<>();
        listFromConfigValuesConfigs.add(
```



```
new Configs()
    .withInputs(listConfigsInputs)
    .withName("fromJobConfig")
);
ConfigValues fromconfigvaluesJobs = new ConfigValues();
fromconfigvaluesJobs.withConfigs(listFromConfigValuesConfigs);
List<Input> listConfigsInputs1 = new ArrayList<>();
listConfigsInputs1.add(
    new Input()
        .withName("retryJobConfig.retryJobType")
        .withValue("NONE")
);
List<Input> listConfigsInputs2 = new ArrayList<>();
listConfigsInputs2.add(
    new Input()
        .withName("schedulerConfig.isSchedulerJob")
        .withValue("false")
);
listConfigsInputs2.add(
    new Input()
        .withName("schedulerConfig.disposableType")
        .withValue("NONE")
);
List<Input> listConfigsInputs3 = new ArrayList<>();
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.numExtractors")
        .withValue("1")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.submitToCluster")
        .withValue("false")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.numLoaders")
        .withValue("1")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.recordDirtyData")
        .withValue("false")
);
List<Configs> listDriverConfigValuesConfigs = new ArrayList<>();
listDriverConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs1)
        .withName("retryJobConfig")
);
ConfigValues driverconfigvaluesJobs = new ConfigValues();
driverconfigvaluesJobs.withConfigs(listDriverConfigValuesConfigs);
List<Input> listConfigsInputs4 = new ArrayList<>();
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.streamName")
        .withValue("dis-lkGm")
);
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.separator")
        .withValue("|")
);
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.columnList")
        .withValue("1&2&3")
);
List<Configs> listToConfigValuesConfigs = new ArrayList<>();
```

```
listToConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs4)
        .withName("toJobConfig")
);
ConfigValues toconfigvaluesJobs = new ConfigValues();
toconfigvaluesJobs.withConfigs(listToConfigValuesConfigs);
List<Job> listbodyJobs = new ArrayList<>();
listbodyJobs.add(
    new Job()
        .withJobType(Job.JobTypeEnum.fromValue("NORMAL_JOB"))
        .withFromConnectorName("elasticsearch-connector")
        .withToConfigValues(toconfigvaluesJobs)
        .withToLinkName("dis")
        .withDriverConfigValues(driverconfigvaluesJobs)
        .withFromConfigValues(fromconfigvaluesJobs)
        .withToConnectorName("dis-connector")
        .withName("es_css")
        .withFromLinkName("css")
);
body.withClusters(listbodyClusters);
body.withJobs(listbodyJobs);
request.withBody(body);
try {
    CreateAndStartRandomClusterJobResponse response =
client.createAndStartRandomClusterJob(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

在CDM集群ID列表中随机选择一个集群，创建一个源端为elasticsearch，目的端为DIS，作业名为es\_css的表迁移作业。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = CreateAndStartRandomClusterJobRequest()
    listClustersbody = [
        "b0791496-e111-4e75-b7ca-9277aeab9297",
        "c2db1191-eb6c-464a-a0d3-b434e6c6df26",
        "c2db1191-eb6c-464a-a0d3-b434e6c6df26"
    ]
    listInputsConfigs = [
        Input(
            name="fromJobConfig.index",
            value="52est"
        ),
        Input(
            name="fromJobConfig.type",
            value="est_array"
        ),
        Input(
            name="fromJobConfig.columnList",
            value="array_f1_int:long&array_f2_text:string&array_f3_object:nested"
        ),
        Input(
            name="fromJobConfig.splitNestedField",
            value="false"
        )
    ]
    listConfigsFromconfigvalues = [
        Configs(
            inputs=listInputsConfigs,
            name="fromJobConfig"
        )
    ]
    fromconfigvaluesJobs = ConfigValues(
        configs=listConfigsFromconfigvalues
    )
    listInputsConfigs1 = [
        Input(
            name="retryJobConfig.retryJobType",
            value="NONE"
        )
    ]
    listInputsConfigs2 = [
        Input(
            name="schedulerConfig.isSchedulerJob",
            value="false"
        ),
        Input(
            name="schedulerConfig.disposableType",
            value="NONE"
        )
    ]
    listInputsConfigs3 = [
        Input(
            name="throttlingConfig.numExtractors",
            value="1"
        ),
        Input(
            name="throttlingConfig.submitToCluster",
            value="false"
        ),
        Input(
            name="throttlingConfig.numLoaders",
            value="1"
        ),
        Input(
            name="throttlingConfig.recordDirtyData",
            value="false"
        )
    ]
```

```
]
listConfigsDriverconfigvalues = [
    Configs(
        inputs=listInputsConfigs1,
        name="retryJobConfig"
    )
]
driverconfigvaluesJobs = ConfigValues(
    configs=listConfigsDriverconfigvalues
)
listInputsConfigs4 = [
    Input(
        name="toJobConfig.streamName",
        value="dis-lkGm"
    ),
    Input(
        name="toJobConfig.separator",
        value="|"
    ),
    Input(
        name="toJobConfig.columnList",
        value="1&2&3"
    )
]
listConfigsToconfigvalues = [
    Configs(
        inputs=listInputsConfigs4,
        name="toJobConfig"
    )
]
toconfigvaluesJobs = ConfigValues(
    configs=listConfigsToconfigvalues
)
listJobsbody = [
    Job(
        job_type="NORMAL_JOB",
        from_connector_name="elasticsearch-connector",
        to_config_values=toconfigvaluesJobs,
        to_link_name="dis",
        driver_config_values=driverconfigvaluesJobs,
        from_config_values=fromconfigvaluesJobs,
        to_connector_name="dis-connector",
        name="es_css",
        from_link_name="css"
    )
]
request.body = CdmRandomCreateAndStartJobJsonReq(
    clusters=listClustersbody,
    jobs=listJobsbody
)
response = client.create_and_start_random_cluster_job(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

在CDM集群ID列表中随机选择一个集群，创建一个源端为elasticsearch，目的端为DIS，作业名为es\_css的表迁移作业。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAndStartRandomClusterJobRequest{}
    var listClustersbody = []string{
        "b0791496-e111-4e75-b7ca-9277aeab9297",
        "c2db1191-eb6c-464a-a0d3-b434e6c6df26",
        "c2db1191-eb6c-464a-a0d3-b434e6c6df26",
    }
    valueInputs := "52est"
    var valueInputsInterface interface{} = valueInputs
    valueInputs1 := "est_array"
    var valueInputsInterface1 interface{} = valueInputs1
    valueInputs2 := "array_f1_int:long&array_f2_text:string&array_f3_object:nested"
    var valueInputsInterface2 interface{} = valueInputs2
    valueInputs3 := "false"
    var valueInputsInterface3 interface{} = valueInputs3
    var listInputsConfigs = []model.Input{
        {
            Name: "fromJobConfig.index",
            Value: &valueInputsInterface,
        },
        {
            Name: "fromJobConfig.type",
            Value: &valueInputsInterface1,
        },
        {
            Name: "fromJobConfig.columnList",
            Value: &valueInputsInterface2,
        },
        {
            Name: "fromJobConfig.splitNestedField",
            Value: &valueInputsInterface3,
        },
    }
    var listConfigsFromConfigValues = []model.Configs{
        {
            Inputs: listInputsConfigs,
            Name: "fromJobConfig",
        },
    }
    fromconfigvaluesJobs := &model.ConfigValues{
        Configs: listConfigsFromConfigValues,
    }
    valueInputs4 := "NONE"
    var valueInputsInterface4 interface{} = valueInputs4
```

```
var listInputsConfigs1 = []model.Input{
    {
        Name: "retryJobConfig.retryJobType",
        Value: &valueInputsInterface4,
    },
}
valueInputs5:= "false"
var valueInputsInterface5 interface{} = valueInputs5
valueInputs6:= "NONE"
var valueInputsInterface6 interface{} = valueInputs6
var listInputsConfigs2 = []model.Input{
    {
        Name: "schedulerConfig.isSchedulerJob",
        Value: &valueInputsInterface5,
    },
    {
        Name: "schedulerConfig.disposableType",
        Value: &valueInputsInterface6,
    },
}
valueInputs7:= "1"
var valueInputsInterface7 interface{} = valueInputs7
valueInputs8:= "false"
var valueInputsInterface8 interface{} = valueInputs8
valueInputs9:= "1"
var valueInputsInterface9 interface{} = valueInputs9
valueInputs10:= "false"
var valueInputsInterface10 interface{} = valueInputs10
var listInputsConfigs3 = []model.Input{
    {
        Name: "throttlingConfig.numExtractors",
        Value: &valueInputsInterface7,
    },
    {
        Name: "throttlingConfig.submitToCluster",
        Value: &valueInputsInterface8,
    },
    {
        Name: "throttlingConfig.numLoaders",
        Value: &valueInputsInterface9,
    },
    {
        Name: "throttlingConfig.recordDirtyData",
        Value: &valueInputsInterface10,
    },
}
var listConfigsDriverConfigValues = []model.Configs{
    {
        Inputs: listInputsConfigs1,
        Name: "retryJobConfig",
    },
}
driverconfigvaluesJobs := &model.ConfigValues{
    Configs: listConfigsDriverConfigValues,
}
valueInputs11:= "dis-lkGm"
var valueInputsInterface11 interface{} = valueInputs11
valueInputs12:= "|"
var valueInputsInterface12 interface{} = valueInputs12
valueInputs13:= "1&2&3"
var valueInputsInterface13 interface{} = valueInputs13
var listInputsConfigs4 = []model.Input{
    {
        Name: "toJobConfig.streamName",
        Value: &valueInputsInterface11,
    },
    {
        Name: "toJobConfig.separator",
        Value: &valueInputsInterface12,
    },
}
```

```

    },
    {
        Name: "toJobConfig.columnList",
        Value: &valueInputsInterface13,
    },
}
var listConfigsToConfigValues = []model.Configs{
    {
        Inputs: listInputsConfigs4,
        Name: "toJobConfig",
    },
}
toconfigvaluesJobs := &model.ConfigValues{
    Configs: listConfigsToConfigValues,
}
var listJobsbody = []model.Job{
    {
        JobType: model.GetJobJobTypeEnum().NORMAL_JOB,
        FromConnectorName: "elasticsearch-connector",
        ToConfigValues: toconfigvaluesJobs,
        ToLinkName: "dis",
        DriverConfigValues: driverconfigvaluesJobs,
        FromConfigValues: fromconfigvaluesJobs,
        ToConnectorName: "dis-connector",
        Name: "es_css",
        FromLinkName: "css",
    },
}
request.Body = &model.CdmRandomCreateAndStartJobJsonReq{
    Clusters: listClustersbody,
    Jobs: listJobsbody,
}
response, err := client.CreateAndStartRandomClusterJob(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。

## 错误码

请参见[错误码](#)。

## 4.2.5 停止作业

### 功能介绍

停止作业接口。

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/job/{job\_name}/stop

表 4-134 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
job_name	是	String	作业名称。

## 请求参数

表 4-135 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 200

表 4-136 响应 Body 参数

参数	参数类型	描述
validation-result	Array of <a href="#">JobValidationResult</a> objects	校验结构：如果停止作业接失败，返回失败原因，请参见validation-result参数说明。如果停止成功，返回空列表。

表 4-137 JobValidationResult

参数	参数类型	描述
message	String	错误描述。



参数	参数类型	描述
status	String	错误级别，如：ERROR、WARNING。

## 请求示例

```
PUT /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/jdbc2hive/stop
```

## 响应示例

**状态码： 200**

OK。

```
{}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class StopJobSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        StopJobRequest request = new StopJobRequest();
        request.withClusterId("{cluster_id}");
        request.withJobName("{job_name}");
        try {
            StopJobResponse response = client.stopJob(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = StopJobRequest()
        request.cluster_id = "{cluster_id}"
        request.job_name = "{job_name}"
        response = client.stop_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cdm.NewCdmClient(
    cdm.CdmClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.StopJobRequest{
    request.ClusterId = "{cluster_id}"
    request.JobName = "{job_name}"
}
response, err := client.StopJob(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。

## 错误码

请参见[错误码](#)。

## 4.2.6 指定集群创建作业

### 功能介绍

指定集群创建作业接口。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/job

表 4-138 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。

## 请求参数

表 4-139 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-140 请求 Body 参数

参数	是否必选	参数类型	描述
jobs	是	Array of <a href="#">Job</a> objects	作业列表，请参见jobs数据结构说明。

表 4-141 Job

参数	是否必选	参数类型	描述
job_type	是	String	作业类型： <ul style="list-style-type: none"> <li>• NORMAL_JOB：表/文件迁移。</li> <li>• BATCH_JOB：整库迁移。</li> <li>• SCENARIO_JOB：场景迁移。</li> </ul>

参数	是否必选	参数类型	描述
from-connector-name	是	String	<p>源端连接类型，对应的连接参数如下：</p> <ul style="list-style-type: none"> <li>• generic-jdbc-connector：关系数据库连接。</li> <li>• obs-connector：OBS连接。</li> <li>• hdfs-connector：HDFS连接。</li> <li>• hbase-connector：HBase连接、CloudTable连接。</li> <li>• hive-connector：Hive连接。</li> <li>• ftp-connector/sftp-connector：FTP/SFTP连接。</li> <li>• mongodb-connector：MongoDB连接。</li> <li>• redis-connector：Redis/DCS连接。</li> <li>• kafka-connector：Kafka连接。</li> <li>• dis-connector：DIS连接。</li> <li>• elasticsearch-connector：Elasticsearch/云搜索服务连接。</li> <li>• dli-connector：DLI连接。</li> <li>• http-connector：HTTP/HTTPS连接，该连接暂无连接参数。</li> <li>• dms-kafka-connector：DMSKafka连接。</li> </ul>
to-config-values	是	<b>ConfigValues</b> object	目的连接参数配置。根据不同目的端有不同的参数配置，具体可参考 <a href="#">目的端作业参数说明</a> 下相应的目的端参数配置。
to-link-name	是	String	目的端连接名称，即为通过“创建连接”接口创建的连接对应的连接名。
driver-config-values	是	<b>ConfigValues</b> object	作业任务参数配置。例如配置作业失败重试、抽取并发数，具体可参考 <a href="#">作业任务参数说明</a> 。

参数	是否必选	参数类型	描述
from-config-values	是	ConfigValues object	源连接参数配置。根据不同源端有不同的参数配置，具体可参考 <a href="#">源端作业参数说明</a> 下相应的源端参数配置。
to-connector-name	是	String	目的端连接类型，对应的连接参数如下： <ul style="list-style-type: none"> <li>• generic-jdbc-connector: 关系数据库连接。</li> <li>• obs-connector: OBS连接。</li> <li>• hdfs-connector: HDFS连接。</li> <li>• hbase-connector: HBase连接、CloudTable连接。</li> <li>• hive-connector: Hive连接。</li> <li>• ftp-connector/sftp-connector: FTP/SFTP连接。</li> <li>• mongodb-connector: MongoDB连接。</li> <li>• redis-connector: Redis/DCS连接。</li> <li>• kafka-connector: Kafka连接。</li> <li>• dis-connector: DIS连接。</li> <li>• elasticsearch-connector: Elasticsearch/云搜索服务连接。</li> <li>• dli-connector: DLI连接。</li> <li>• http-connector: HTTP/HTTPS连接，该连接暂无连接参数。</li> <li>• dms-kafka-connector: DMSKafka连接。</li> </ul>
name	是	String	作业名称，长度在1到240个字符之间。
from-link-name	是	String	源连接名称，即为通过“创建连接”接口创建的连接对应的连接名。
creation-user	否	String	创建作业的用户。由系统生成，用户无需填写。
creation-date	否	Long	作业创建的时间，单位：毫秒。由系统生成，用户无需填写。

参数	是否必选	参数类型	描述
update-date	否	Long	作业最后更新的时间，单位：毫秒。由系统生成，用户无需填写。
is_incre_job	否	Boolean	是否是增量作业。已废弃。
flag	否	Integer	是否是定时作业标记，如果是定时作业则为1，否则为0。由系统根据定时任务配置生成，用户无需填写。
files_read	否	Integer	已读文件数。由系统生成，用户无需填写。
update-user	否	String	最后更新作业的用户。由系统生成，用户无需填写。
external_id	否	String	具体执行的作业id，如果是本地作业，则一般为"job_local1202051771_0002"形式，如果是DLI作业，则为DLI作业ID，比如"12345"。由系统生成，用户无需填写。
type	否	String	与job_type一致，作业类型： <ul style="list-style-type: none"> <li>• NORMAL_JOB：表/文件迁移。</li> <li>• BATCH_JOB：整库迁移。</li> <li>• SCENARIO_JOB：场景迁移。</li> </ul>
execute_start_date	否	Long	最近一次执行任务开始时间，单位：毫秒。由系统生成，用户无需填写。
delete_rows	否	Integer	增量作业删除行数，已废弃。
enabled	否	Boolean	是否激活连接。由系统生成，用户无需填写。
bytes_written	否	Long	作业写入的字节。由系统生成，用户无需填写。
id	否	Integer	作业ID。由系统生成，用户无需填写。
is_use_sql	否	Boolean	用户是否使用sql。由系统根据源端抽取是否使用sql语句生成，用户无需填写。
update_rows	否	Integer	增量作业更新行数，已废弃。
group_name	否	String	组名。

参数	是否必选	参数类型	描述
bytes_read	否	Long	作业读取的字节。由系统生成，用户无需填写。
execute_update_date	否	Long	最近一次执行任务更新时间，单位：毫秒。由系统生成，用户无需填写。
write_rows	否	Integer	增量作业写入行数，已废弃。
rows_written	否	Integer	作业写入的行数。由系统生成，用户无需填写。
rows_read	否	Long	作业读取的行数。由系统生成，用户无需填写。
files_written	否	Integer	写入文件数。由系统生成，用户无需填写。
is_incrementing	否	Boolean	是否是增量作业，同 is_incre_job，已废弃。
execute_create_date	否	Long	最近一次执行任务创建时间，单位：毫秒。由系统生成，用户无需填写。
status	否	String	作业最后的执行状态： <ul style="list-style-type: none"> <li>• BOOTING：启动中。</li> <li>• RUNNING：运行中。</li> <li>• SUCCEEDED：成功。</li> <li>• FAILED：失败。</li> <li>• NEW：未被执行。</li> </ul>

表 4-142 ConfigValues

参数	是否必选	参数类型	描述
configs	是	Array of <b>configs</b> objects	源连接参数、目的连接参数和作业任务参数，它们的配置数据结构相同，其中“inputs”里的参数不一样，详细请参见configs数据结构说明。
extended-configs	否	<b>extended-configs</b> object	扩展配置，请参见extended-configs参数说明。扩展配置暂不对外开放，用户无需填写。



表 4-143 configs

参数	是否必选	参数类型	描述
inputs	是	Array of <b>Input</b> objects	输入参数列表，列表中的每个参数为“name,value”结构，请参考inputs数据结构参数说明。在“from-config-values”数据结构中，不同的源连接类型有不同的“inputs”参数列表，请参见源端作业参数说明下的章节。在“to-config-values”数据结构中，不同的目的连接类型有不同的“inputs”参数列表，请参见目的端作业参数说明下面的子章节。在“driver-config-values”数据结构中，“inputs”具体参数请参见作业任务参数说明。
name	是	String	配置名称：源端作业的配置名称为“fromJobConfig”。目的端作业的配置名称为“toJobConfig”，连接的配置名称固定为“linkConfig”。
id	否	Integer	配置ID，由系统生成，用户无需填写。
type	否	String	配置类型，由系统生成，用户无需填写。值为LINK或者JOB，如果是连接管理API，则为LINK；如果是作业管理API，则为JOB。

表 4-144 Input

参数	是否必选	参数类型	描述
name	是	String	<p>参数名:</p> <ul style="list-style-type: none"> <li>如果是连接管理API, 则以“linkConfig.”开头, 对于不同连接类型有不同的参数, 具体可参见<a href="#">连接参数说明</a>下相应连接的参数说明。</li> <li>如果是作业管理API, 对于源端连接参数, 则以“fromJobConfig.”开头, 具体可参见<a href="#">源端作业参数说明</a>下相应的源端参数说明; 对于目的端连接参数, 则以“toJobConfig.”开头, 具体可参见<a href="#">目的端作业参数说明</a>下相应的目的端参数说明; 对于作业任务参数, 请参见<a href="#">作业任务参数说明</a>下相应的任务参数说明。</li> </ul>
value	是	Object	参数值, 参数名对应的值, 必须填写为字符串。
type	否	String	值类型, 如STRING、INTEGER, 由系统设定, 用户无需填写。

表 4-145 extended-configs

参数	是否必选	参数类型	描述
name	否	String	扩展配置名称, 暂不对外开放, 用户无需填写。
value	否	String	扩展配置值, 暂不对外开放, 用户无需填写。

## 响应参数

状态码: 200

表 4-146 响应 Body 参数

参数	参数类型	描述
name	String	作业名称。

参数	参数类型	描述
validation-result	Array of <b>JobValidationResult</b> objects	校验结果： <ul style="list-style-type: none"> <li>如果修改失败，返回失败原因。</li> <li>如果修改成功，返回空列表。</li> </ul>

表 4-147 JobValidationResult

参数	参数类型	描述
message	String	错误描述。
status	String	错误级别，如：ERROR、WARNING。

状态码： 400

表 4-148 响应 Body 参数

参数	参数类型	描述
code	String	返回编码。
errCode	String	错误码。
message	String	报错信息。
externalMessage	String	附加信息。

## 请求示例

创建一个源端为Elasticsearch数据连接，目的端为DIS数据连接，作业名为es\_css的数据迁移作业。

```
POST /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job
{
  "jobs": [ {
    "job_type": "NORMAL_JOB",
    "from-connector-name": "elasticsearch-connector",
    "to-config-values": {
      "configs": [ {
        "inputs": [ {
          "name": "toJobConfig.streamName",
          "value": "dis-lkGm"
        }, {
          "name": "toJobConfig.separator",
          "value": "|"
        }, {
          "name": "toJobConfig.columnList",
          "value": "1&2&3"
        }
      ],
      "name": "toJobConfig"
    }
  }
]
```

```

    }]
  },
  "to-link-name": "dis",
  "driver-config-values": {
    "configs": [ {
      "inputs": [ {
        "name": "throttlingConfig.numExtractors",
        "value": "1"
      }, {
        "name": "throttlingConfig.submitToCluster",
        "value": "false"
      }, {
        "name": "throttlingConfig.numLoaders",
        "value": "1"
      }, {
        "name": "throttlingConfig.recordDirtyData",
        "value": "false"
      }
    ],
    "name": "throttlingConfig"
  }, {
    "inputs": [ ],
    "name": "jarConfig"
  }, {
    "inputs": [ {
      "name": "schedulerConfig.isSchedulerJob",
      "value": "false"
    }, {
      "name": "schedulerConfig.disposableType",
      "value": "NONE"
    }
  ],
  "name": "schedulerConfig"
  }, {
    "inputs": [ ],
    "name": "transformConfig"
  }, {
    "inputs": [ {
      "name": "retryJobConfig.retryJobType",
      "value": "NONE"
    }
  ],
  "name": "retryJobConfig"
  }
  }],
  "from-config-values": {
    "configs": [ {
      "inputs": [ {
        "name": "fromJobConfig.index",
        "value": "52est"
      }, {
        "name": "fromJobConfig.type",
        "value": "est_array"
      }, {
        "name": "fromJobConfig.columnList",
        "value": "array_f1_int:long&array_f2_text:string&array_f3_object:nested"
      }, {
        "name": "fromJobConfig.splitNestedField",
        "value": "false"
      }
    ],
    "name": "fromJobConfig"
  }
  }],
  "to-connector-name": "dis-connector",
  "name": "es_css",
  "from-link-name": "css"
}
}

```

## 响应示例

状态码: 200

OK。

```
{  
  "name" : "mysql2hive"  
}
```

**状态码： 400**

请求报错。

```
{  
  "code" : "Cdm.0104",  
  "errCode" : "Cdm.0104",  
  "message" : "Job name already exist or created by other.",  
  "ternalMessage" : "Job name already exist or created by other."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个源端为Elasticsearch数据连接，目的端为DIS数据连接，作业名为es\_css的数据迁移作业。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;  
import com.huaweicloud.sdk.cdm.v1.*;  
import com.huaweicloud.sdk.cdm.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CreateJobSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CdmClient client = CdmClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateJobRequest request = new CreateJobRequest();  
        request.withClusterId("{cluster_id}");  
        CdmCreateJobJsonReq body = new CdmCreateJobJsonReq();  
        List<Input> listConfigsInputs = new ArrayList<>();  
        listConfigsInputs.add(  
            new Input()
```

```
.withName("fromJobConfig.index")
.withValue("52est")
);
listConfigsInputs.add(
    new Input()
        .withName("fromJobConfig.type")
        .withValue("est_array")
);
listConfigsInputs.add(
    new Input()
        .withName("fromJobConfig.columnList")
        .withValue("array_f1_int:long&array_f2_text:string&array_f3_object:nested")
);
listConfigsInputs.add(
    new Input()
        .withName("fromJobConfig.splitNestedField")
        .withValue("false")
);
List<Configs> listFromConfigValuesConfigs = new ArrayList<>();
listFromConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs)
        .withName("fromJobConfig")
);
ConfigValues fromconfigvaluesJobs = new ConfigValues();
fromconfigvaluesJobs.withConfigs(listFromConfigValuesConfigs);
List<Input> listConfigsInputs1 = new ArrayList<>();
listConfigsInputs1.add(
    new Input()
        .withName("retryJobConfig.retryJobType")
        .withValue("NONE")
);
List<Input> listConfigsInputs2 = new ArrayList<>();
listConfigsInputs2.add(
    new Input()
        .withName("schedulerConfig.isSchedulerJob")
        .withValue("false")
);
listConfigsInputs2.add(
    new Input()
        .withName("schedulerConfig.disposableType")
        .withValue("NONE")
);
List<Input> listConfigsInputs3 = new ArrayList<>();
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.numExtractors")
        .withValue("1")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.submitToCluster")
        .withValue("false")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.numLoaders")
        .withValue("1")
);
listConfigsInputs3.add(
    new Input()
        .withName("throttlingConfig.recordDirtyData")
        .withValue("false")
);
List<Configs> listDriverConfigValuesConfigs = new ArrayList<>();
listDriverConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs1)
        .withName("retryJobConfig")
);
```

```
);
ConfigValues driverconfigvaluesJobs = new ConfigValues();
driverconfigvaluesJobs.withConfigs(listDriverConfigValuesConfigs);
List<Input> listConfigsInputs4 = new ArrayList<>();
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.streamName")
        .withValue("dis-lkGm")
);
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.separator")
        .withValue("|")
);
listConfigsInputs4.add(
    new Input()
        .withName("toJobConfig.columnList")
        .withValue("1&2&3")
);
List<Configs> listToConfigValuesConfigs = new ArrayList<>();
listToConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs4)
        .withName("toJobConfig")
);
ConfigValues toconfigvaluesJobs = new ConfigValues();
toconfigvaluesJobs.withConfigs(listToConfigValuesConfigs);
List<Job> listbodyJobs = new ArrayList<>();
listbodyJobs.add(
    new Job()
        .withJobType(Job.JobTypeEnum.fromValue("NORMAL_JOB"))
        .withFromConnectorName("elasticsearch-connector")
        .withToConfigValues(toconfigvaluesJobs)
        .withToLinkName("dis")
        .withDriverConfigValues(driverconfigvaluesJobs)
        .withFromConfigValues(fromconfigvaluesJobs)
        .withToConnectorName("dis-connector")
        .withName("es_css")
        .withFromLinkName("css")
);
body.withJobs(listbodyJobs);
request.withBody(body);
try {
    CreateJobResponse response = client.createJob(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

创建一个源端为Elasticsearch数据连接，目的端为DIS数据连接，作业名为es\_css的数据迁移作业。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateJobRequest()
        request.cluster_id = "{cluster_id}"
        listInputsConfigs = [
            Input(
                name="fromJobConfig.index",
                value="52est"
            ),
            Input(
                name="fromJobConfig.type",
                value="est_array"
            ),
            Input(
                name="fromJobConfig.columnList",
                value="array_f1_int:long&array_f2_text:string&array_f3_object:nested"
            ),
            Input(
                name="fromJobConfig.splitNestedField",
                value="false"
            )
        ]
        listConfigsFromconfigvalues = [
            Configs(
                inputs=listInputsConfigs,
                name="fromJobConfig"
            )
        ]
        fromconfigvaluesJobs = ConfigValues(
            configs=listConfigsFromconfigvalues
        )
        listInputsConfigs1 = [
            Input(
                name="retryJobConfig.retryJobType",
                value="NONE"
            )
        ]
        listInputsConfigs2 = [
            Input(
                name="schedulerConfig.isSchedulerJob",
                value="false"
            ),
            Input(
                name="schedulerConfig.disposableType",
                value="NONE"
            )
        ]
        listInputsConfigs3 = [
            Input(
                name="throttlingConfig.numExtractors",
```



```

        value="1"
    ),
    Input(
        name="throttlingConfig.submitToCluster",
        value="false"
    ),
    Input(
        name="throttlingConfig.numLoaders",
        value="1"
    ),
    Input(
        name="throttlingConfig.recordDirtyData",
        value="false"
    )
]
listConfigsDriverconfigvalues = [
    Configs(
        inputs=listInputsConfigs1,
        name="retryJobConfig"
    )
]
driverconfigvaluesJobs = ConfigValues(
    configs=listConfigsDriverconfigvalues
)
listInputsConfigs4 = [
    Input(
        name="toJobConfig.streamName",
        value="dis-lkGm"
    ),
    Input(
        name="toJobConfig.separator",
        value="|"
    ),
    Input(
        name="toJobConfig.columnList",
        value="1&2&3"
    )
]
listConfigsToconfigvalues = [
    Configs(
        inputs=listInputsConfigs4,
        name="toJobConfig"
    )
]
toconfigvaluesJobs = ConfigValues(
    configs=listConfigsToconfigvalues
)
listJobsbody = [
    Job(
        job_type="NORMAL_JOB",
        from_connector_name="elasticsearch-connector",
        to_config_values=toconfigvaluesJobs,
        to_link_name="dis",
        driver_config_values=driverconfigvaluesJobs,
        from_config_values=fromconfigvaluesJobs,
        to_connector_name="dis-connector",
        name="es_css",
        from_link_name="css"
    )
]
request.body = CdmCreateJobJsonReq(
    jobs=listJobsbody
)
response = client.create_job(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)

```

```
print(e.error_code)
print(e.error_msg)
```

## Go

创建一个源端为Elasticsearch数据连接，目的端为DIS数据连接，作业名为es\_css的数据迁移作业。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateJobRequest{}
    request.ClusterId = "{cluster_id}"
    valueInputs := "52est"
    var valueInputsInterface interface{} = valueInputs
    valueInputs1 := "est_array"
    var valueInputsInterface1 interface{} = valueInputs1
    valueInputs2 := "array_f1_int:long&array_f2_text:string&array_f3_object:nested"
    var valueInputsInterface2 interface{} = valueInputs2
    valueInputs3 := "false"
    var valueInputsInterface3 interface{} = valueInputs3
    var listInputsConfigs = []model.Input{
        {
            Name: "fromJobConfig.index",
            Value: &valueInputsInterface,
        },
        {
            Name: "fromJobConfig.type",
            Value: &valueInputsInterface1,
        },
        {
            Name: "fromJobConfig.columnList",
            Value: &valueInputsInterface2,
        },
        {
            Name: "fromJobConfig.splitNestedField",
            Value: &valueInputsInterface3,
        },
    }
    var listConfigsFromConfigValues = []model.Configs{
```

```

    {
      Inputs: listInputsConfigs,
      Name: "fromJobConfig",
    },
  },
}
fromconfigvaluesJobs := &model.ConfigValues{
  Configs: listConfigsFromConfigValues,
}
valueInputs4:= "NONE"
var valueInputsInterface4 interface{} = valueInputs4
var listInputsConfigs1 = []model.Input{
  {
    Name: "retryJobConfig.retryJobType",
    Value: &valueInputsInterface4,
  },
}
valueInputs5:= "false"
var valueInputsInterface5 interface{} = valueInputs5
valueInputs6:= "NONE"
var valueInputsInterface6 interface{} = valueInputs6
var listInputsConfigs2 = []model.Input{
  {
    Name: "schedulerConfig.isSchedulerJob",
    Value: &valueInputsInterface5,
  },
  {
    Name: "schedulerConfig.disposableType",
    Value: &valueInputsInterface6,
  },
}
valueInputs7:= "1"
var valueInputsInterface7 interface{} = valueInputs7
valueInputs8:= "false"
var valueInputsInterface8 interface{} = valueInputs8
valueInputs9:= "1"
var valueInputsInterface9 interface{} = valueInputs9
valueInputs10:= "false"
var valueInputsInterface10 interface{} = valueInputs10
var listInputsConfigs3 = []model.Input{
  {
    Name: "throttlingConfig.numExtractors",
    Value: &valueInputsInterface7,
  },
  {
    Name: "throttlingConfig.submitToCluster",
    Value: &valueInputsInterface8,
  },
  {
    Name: "throttlingConfig.numLoaders",
    Value: &valueInputsInterface9,
  },
  {
    Name: "throttlingConfig.recordDirtyData",
    Value: &valueInputsInterface10,
  },
}
}
var listConfigsDriverConfigValues = []model.Configs{
  {
    Inputs: listInputsConfigs1,
    Name: "retryJobConfig",
  },
}
}
driverconfigvaluesJobs := &model.ConfigValues{
  Configs: listConfigsDriverConfigValues,
}
valueInputs11:= "dis-lkGm"
var valueInputsInterface11 interface{} = valueInputs11
valueInputs12:= "|"
var valueInputsInterface12 interface{} = valueInputs12

```

```

valueInputs13:= "1&2&3"
var valueInputsInterface13 interface{} = valueInputs13
var listInputsConfigs4 = []model.Input{
    {
        Name: "toJobConfig.streamName",
        Value: &valueInputsInterface11,
    },
    {
        Name: "toJobConfig.separator",
        Value: &valueInputsInterface12,
    },
    {
        Name: "toJobConfig.columnList",
        Value: &valueInputsInterface13,
    },
}
var listConfigsToConfigValues = []model.Configs{
    {
        Inputs: listInputsConfigs4,
        Name: "toJobConfig",
    },
}
toconfigvaluesJobs := &model.ConfigValues{
    Configs: listConfigsToConfigValues,
}
var listJobsbody = []model.Job{
    {
        JobType: model.GetJobJobTypeEnum().NORMAL_JOB,
        FromConnectorName: "elasticsearch-connector",
        ToConfigValues: toconfigvaluesJobs,
        ToLinkName: "dis",
        DriverConfigValues: driverconfigvaluesJobs,
        FromConfigValues: fromconfigvaluesJobs,
        ToConnectorName: "dis-connector",
        Name: "es_css",
        FromLinkName: "css",
    },
}
request.Body = &model.CdmCreateJobJsonReq{
    Jobs: listJobsbody,
}
response, err := client.CreateJob(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求报错。

## 错误码

请参见[错误码](#)。

## 4.2.7 启动作业

### 功能介绍

启动作业接口。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/job/{job\_name}/start

表 4-149 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
job_name	是	String	作业名称。

### 请求参数

表 4-150 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-151 请求 Body 参数

参数	是否必选	参数类型	描述
variables	否	Object	启动作业，配置变量参数，作业配置无变量时，为空对象。

## 响应参数

状态码： 200

表 4-152 响应 Body 参数

参数	参数类型	描述
submissions	Array of <a href="#">StartJobSubmission</a> objects	作业运行信息，请参见 <a href="#">submission</a> 参数说明。

表 4-153 StartJobSubmission

参数	参数类型	描述
isIncrementing	Boolean	作业是否为增量迁移。
delete_rows	Integer	删除数据行数。
update_rows	Integer	更新数据行数。
write_rows	Integer	写入数据行数。
submission-id	Integer	作业提交id。
job-name	String	作业名称。
creation-user	String	创建用户。
creation-date	Long	创建时间，单位：毫秒。
execute-date	Long	执行时间。
progress	Float	作业进度，失败时为“-1”，其它情况为0~100。
status	String	作业状态： <ul style="list-style-type: none"> <li>● BOOTING：启动中。</li> <li>● FAILURE_ON_SUBMIT：提交失败。</li> <li>● RUNNING：运行中。</li> <li>● SUCCEEDED：成功。</li> <li>● FAILED：失败。</li> <li>● UNKNOWN：未知。</li> <li>● NEVER_EXECUTED：未被执行。</li> </ul>
isStoppingIncrement	String	是否停止增量迁移。

参数	参数类型	描述
is-execute-auto	Boolean	是否定时执行作业。
last-update-date	Long	作业最后更新时间。
last-udpate-user	String	最后更新作业状态的用户。
isDeleteJob	Boolean	作业执行完成后是否删除。

## 请求示例

启动作业，参数为空。

```
PUT /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/jdbc2hive/start
{
  "variables" : {}
}
```

## 响应示例

状态码： 200

OK。

```
{
  "submissions" : [{
    "job-name" : "jdbc2hive",
    "creation-user" : "cdm",
    "creation-date" : "1536905778725",
    "progress" : 1,
    "status" : "BOOTING"
  }]
}
```

## 状态码

状态码	描述
200	OK。

## 错误码

请参见[错误码](#)。

## 4.2.8 查询作业状态

### 功能介绍

查询作业状态接口。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/job/{job\_name}/status

表 4-154 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
job_name	是	String	作业名称。

## 请求参数

表 4-155 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 200

表 4-156 响应 Body 参数

参数	参数类型	描述
submissions	Array of <a href="#">Submission</a> objects	作业运行信息，详见submissions参数说明。

表 4-157 Submission

参数	参数类型	描述
isIncrementing	Boolean	作业是否为增量迁移。



参数	参数类型	描述
job-name	String	作业名称。
counters	<b>counters</b> object	作业运行结果统计，当“status”为“SUCCEEDED”时才有此字段，请参见counters数据结构参数说明。
isStopingIncrement	String	是否停止增量迁移。
is-execute-auto	Boolean	是否定时执行作业。
last-update-date	Long	作业最后更新时间。
last-udpate-user	String	最后更新作业状态的用户。
isDeleteJob	Boolean	作业执行完成后是否删除。
creation-user	String	创建用户。
creation-date	Long	创建时间。
external-id	String	作业ID。
progress	Float	作业进度，失败时为“-1”，其它情况为0~100。
submission-id	Integer	作业提交id。
delete_rows	Integer	删除数据行数。
update_rows	Integer	更新数据行数。
write_rows	Integer	写入数据行数。
execute-date	Long	执行时间。
status	String	作业状态： <ul style="list-style-type: none"> <li>● BOOTING：启动中。</li> <li>● FAILURE_ON_SUBMIT：提交失败。</li> <li>● RUNNING：运行中。</li> <li>● SUCCEEDED：成功。</li> <li>● FAILED：失败。</li> <li>● UNKNOWN：未知。</li> <li>● NEVER_EXECUTED：未被执行。</li> </ul>
error-details	String	错误详情，当“status”为“FAILED”时才有此字段。
error-summary	String	错误总结，当“status”为“FAILED”时才有此字段。

表 4-158 counters

参数	参数类型	描述
org.apache.hadoop.submission.counter.SqoopCounters	counter object	作业运行结果统计，请参见统计结果参数说明。

表 4-159 counter

参数	参数类型	描述
BYTES_WRITTEN	Long	写入的字节数。
TOTAL_FILES	Integer	总文件数。
ROWS_READ	Long	读取的行数。
BYTES_READ	Long	读取的字节数。
ROWS_WRITTEN	Long	写入的行数。
FILES_WRITTEN	Integer	写入的文件数。
FILES_READ	Integer	读取的文件数。
TOTAL_SIZE	Long	总字节数。
FILES_SKIPPED	Integer	跳过的文件数。
ROWS_WRITTEN_SKIPPED	Long	跳过的行数。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/jdbc2hive/status
```

## 响应示例

状态码： 200

OK。

```
{
  "submissions": [ {
    "job-name": "jdbc2hive",
    "creation-user": "cdm",
```

```
"creation-date": "1536905778725",  
"progress": 1,  
"status": "BOOTING"  
}]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;  
import com.huaweicloud.sdk.cdm.v1.*;  
import com.huaweicloud.sdk.cdm.v1.model.*;  
  
public class ShowJobStatusSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CdmClient client = CdmClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowJobStatusRequest request = new ShowJobStatusRequest();  
        request.withClusterId("{cluster_id}");  
        request.withJobName("{job_name}");  
        try {  
            ShowJobStatusResponse response = client.showJobStatus(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowJobStatusRequest()
        request.cluster_id = "{cluster_id}"
        request.job_name = "{job_name}"
        response = client.show_job_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
```

```

        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build()

    request := &model.ShowJobStatusRequest{
    request.ClusterId = "{cluster_id}"
    request.JobName = "{job_name}"
    response, err := client.ShowJobStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。

## 错误码

请参见[错误码](#)。

## 4.2.9 查询作业执行历史

### 功能介绍

查询作业执行历史接口。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/submissions

表 4-160 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。

表 4-161 Query 参数

参数	是否必选	参数类型	描述
jname	是	String	作业名称。

## 请求参数

表 4-162 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 200

表 4-163 响应 Body 参数

参数	参数类型	描述
submissions	Array of <b>Submission</b> objects	作业运行信息，详见submissions参数说明。
total	Integer	查询该作业总的历史记录数。
page_no	Integer	查询作业记录时，分页数。
page_size	Integer	分页查询，每页返回的记录数。默认值：10。

表 4-164 Submission

参数	参数类型	描述
isIncrementing	Boolean	作业是否为增量迁移。
job-name	String	作业名称。
counters	<b>counters</b> object	作业运行结果统计，当“status”为“SUCCEEDED”时才有此字段，请参见counters数据结构参数说明。

参数	参数类型	描述
isStopingIncrement	String	是否停止增量迁移。
is-execute-auto	Boolean	是否定时执行作业。
last-update-date	Long	作业最后更新时间。
last-udpate-user	String	最后更新作业状态的用户。
isDeleteJob	Boolean	作业执行完成后是否删除。
creation-user	String	创建用户。
creation-date	Long	创建时间。
external-id	String	作业ID。
progress	Float	作业进度，失败时为“-1”，其它情况为0~100。
submission-id	Integer	作业提交id。
delete_rows	Integer	删除数据行数。
update_rows	Integer	更新数据行数。
write_rows	Integer	写入数据行数。
execute-date	Long	执行时间。
status	String	作业状态： <ul style="list-style-type: none"> <li>● BOOTING：启动中。</li> <li>● FAILURE_ON_SUBMIT：提交失败。</li> <li>● RUNNING：运行中。</li> <li>● SUCCEEDED：成功。</li> <li>● FAILED：失败。</li> <li>● UNKNOWN：未知。</li> <li>● NEVER_EXECUTED：未被执行。</li> </ul>
error-details	String	错误详情，当“status”为“FAILED”时才有此字段。
error-summary	String	错误总结，当“status”为“FAILED”时才有此字段。

表 4-165 counters

参数	参数类型	描述
org.apache.hadoop.submission.counter.SqoopCounters	counter object	作业运行结果统计，请参见统计结果参数说明。

表 4-166 counter

参数	参数类型	描述
BYTES_WRITTEN	Long	写入的字节数。
TOTAL_FILES	Integer	总文件数。
ROWS_READ	Long	读取的行数。
BYTES_READ	Long	读取的字节数。
ROWS_WRITTEN	Long	写入的行数。
FILES_WRITTEN	Integer	写入的文件数。
FILES_READ	Integer	读取的文件数。
TOTAL_SIZE	Long	总字节数。
FILES_SKIPPED	Integer	跳过的文件数。
ROWS_WRITTEN_SKIPPED	Long	跳过的行数。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/submissions?jname=jdbc2hive
```

## 响应示例

状态码： 200

OK。

```
{
  "submissions": [ {
    "job-name": "jdbc2hive",
    "creation-user": "cdm",
    "creation-date": "1536905778725",
    "progress": 1,
    "status": "BOOTING"
  }
]
```



```
    }]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;  
import com.huaweicloud.sdk.cdm.v1.*;  
import com.huaweicloud.sdk.cdm.v1.model.*;  
  
public class ShowSubmissionsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CdmClient client = CdmClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowSubmissionsRequest request = new ShowSubmissionsRequest();  
        request.withClusterId("{cluster_id}");  
        try {  
            ShowSubmissionsResponse response = client.showSubmissions(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSubmissionsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_submissions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSubmissionsRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowSubmissions(request)
```

```
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。

## 错误码

请参见[错误码](#)。

# 4.3 连接管理

## 4.3.1 创建连接

### 功能介绍

创建连接接口。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/link

表 4-167 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。

表 4-168 Query 参数

参数	是否必选	参数类型	描述
validate	否	String	为“true”时，此API仅校验参数是否正确，不创建连接。

## 请求参数

表 4-169 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-170 请求 Body 参数

参数	是否必选	参数类型	描述
links	是	Array of <a href="#">links</a> objects	连接列表，请参见links数据结构说明。

表 4-171 links

参数	是否必选	参数类型	描述
link-config-values	是	<a href="#">link-config-values</a> object	连接参数配置，请参见link-config-values参数说明。
creation-user	否	String	创建连接的用户。
name	是	String	连接名称。
id	否	Integer	连接ID。
creation-date	否	Long	创建连接的时间。

参数	是否必选	参数类型	描述
connector-name	是	String	连接器名称，对应的连接参数如下： generic-jdbc-connector: 关系数据库连接。 obs-connector: OBS连接。 hdfs-connector: HDFS连接。 hbase-connector: HBase连接、CloudTable连接。 hive-connector: Hive连接。 ftp-connector/sftp-connector: FTP/SFTP连接。 mongodb-connector: MongoDB连接。 redis-connector: Redis/DCS连接。 kafka-connector: Kafka连接。 dis-connector: DIS连接。 elasticsearch-connector: Elasticsearch/云搜索服务连接。 dli-connector: DLI连接。 http-connector: HTTP/HTTPS连接，该连接暂无连接参数。 dms-kafka-connector: DMSKafka连接。
update-date	否	Long	更新连接的时间。
enabled	否	Boolean	是否激活连接，默认为“true”。
update-user	否	String	更新连接的用户。

表 4-172 link-config-values

参数	是否必选	参数类型	描述
configs	是	Array of <b>configs</b> objects	连接配置参数数据结构，请参见configs参数说明。
extended-configs	否	<b>extended-configs</b> object	扩展配置，请参见extended-configs参数说明。
validators	否	Array of strings	校验器。

表 4-173 configs

参数	是否必选	参数类型	描述
inputs	是	Array of <b>Input</b> objects	输入参数列表，列表中的每个参数为“name,value”结构，请参考inputs数据结构参数说明。在“from-config-values”数据结构中，不同的源连接类型有不同的“inputs”参数列表，请参见源端作业参数说明下的章节。在“to-config-values”数据结构中，不同的目的连接类型有不同的“inputs”参数列表，请参见目的端作业参数说明下面的子章节。在“driver-config-values”数据结构中，“inputs”具体参数请参见作业任务参数说明。
name	是	String	配置名称：源端作业的配置名称为“fromJobConfig”。目的端作业的配置名称为“toJobConfig”，连接的配置名称固定为“linkConfig”。
id	否	Integer	配置ID，由系统生成，用户无需填写。
type	否	String	配置类型，由系统生成，用户无需填写。值为LINK或者JOB，如果是连接管理API，则为LINK；如果是作业管理API，则为JOB。

表 4-174 Input

参数	是否必选	参数类型	描述
name	是	String	<p>参数名:</p> <ul style="list-style-type: none"> <li>如果是连接管理API, 则以“linkConfig.”开头, 对于不同连接类型有不同的参数, 具体可参见<a href="#">连接参数说明</a>下相应连接的参数说明。</li> <li>如果是作业管理API, 对于源端连接参数, 则以“fromJobConfig.”开头, 具体可参见<a href="#">源端作业参数说明</a>下相应的源端参数说明; 对于目的端连接参数, 则以“toJobConfig.”开头, 具体可参见<a href="#">目的端作业参数说明</a>下相应的目的端参数说明; 对于作业任务参数, 请参见<a href="#">作业任务参数说明</a>下相应的任务参数说明。</li> </ul>
value	是	Object	参数值, 参数名对应的值, 必须填写为字符串。
type	否	String	值类型, 如STRING、INTEGER, 由系统设定, 用户无需填写。

表 4-175 extended-configs

参数	是否必选	参数类型	描述
name	否	String	名称。
value	否	String	值。

## 响应参数

状态码: 200

表 4-176 响应 Body 参数

参数	参数类型	描述
name	String	连接名称。

参数	参数类型	描述
validation-result	Array of <b>validationResult</b> objects	校验结构：如果创建连接失败，返回失败原因，请参见validation-result参数说明。如果创建成功，返回空列表。

表 4-177 validationResult

参数	参数类型	描述
linkConfig	Array of <b>validationLinkConfig</b> objects	创建或更新连接校验结果，请参见linkConfig参数说明。

表 4-178 validationLinkConfig

参数	参数类型	描述
message	String	错误描述。
status	String	错误级别，如：ERROR、WARNING。

#### 状态码： 400

表 4-179 响应 Body 参数

参数	参数类型	描述
code	String	返回编码。
errCode	String	错误码。
message	String	报错信息。
externalMessage	String	附加信息。

#### 状态码： 500

表 4-180 响应 Body 参数

参数	参数类型	描述
message	String	错误描述。
status	String	错误级别，如：ERROR、WARNING。



## 请求示例

创建一个名为mysql\_link的数据连接。

```
POST /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/link
{
  "links": [ {
    "link-config-values": {
      "configs": [ {
        "inputs": [ {
          "name": "linkConfig.databaseType",
          "value": "MYSQL"
        }, {
          "name": "linkConfig.host",
          "value": "100.94.8.163"
        }, {
          "name": "linkConfig.port",
          "value": "3306"
        }, {
          "name": "linkConfig.database",
          "value": "DB_name"
        }, {
          "name": "linkConfig.username",
          "value": "username"
        }, {
          "name": "linkConfig.password",
          "value": "DB_password"
        }, {
          "name": "linkConfig.fetchSize",
          "value": "100000"
        }, {
          "name": "linkConfig.usingNative",
          "value": "false"
        }
      ],
      "name": "linkConfig"
    }
  ],
  "name": "mysql_link",
  "creation-date": 1496654788622,
  "connector-name": "generic-jdbc-connector",
  "update-date": 1496654788622,
  "enabled": true
}
}
```

## 响应示例

**状态码： 200**

OK。

```
{
  "name": "rdb_link",
  "validation-result": [ { } ]
}
```

**状态码： 400**

请求错误。

```
{
  "code": "Cdm.0315",
  "errCode": "Cdm.0315",
  "message": "Link name [ftp_link] already exist or created by other user.",
  "externalMessage": "Link name [ftp_link] already exist or created by other user."
}
```

**状态码： 500**

服务内部错误，具体返回错误码请参考错误码。

```
{
  "validation-result": [ {
    "linkConfig": [ {
      "message": "Can't connect to the database with given credentials: The authentication type 12 is not supported. Check that you have configured the pg_hba.conf file to include the client's IP address or subnet, and that it is using an authentication scheme supported by the driver.",
      "status": "ERROR"
    } ]
  } ]
}
```

**SDK 代码示例**

SDK代码示例如下。

**Java**

创建一个名为mysql\_link的数据连接。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateLinkSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateLinkRequest request = new CreateLinkRequest();
        request.withClusterId("{cluster_id}");
        CdmCreateAndUpdateLinkReq body = new CdmCreateAndUpdateLinkReq();
        List<Input> listConfigsInputs = new ArrayList<>();
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.databaseType")
                .withValue("MYSQL")
        );
    }
}
```

```
listConfigsInputs.add(
    new Input()
        .withName("linkConfig.host")
        .withValue("100.94.8.163")
);
listConfigsInputs.add(
    new Input()
        .withName("linkConfig.port")
        .withValue("3306")
);
listConfigsInputs.add(
    new Input()
        .withName("linkConfig.database")
        .withValue("DB_name")
);
listConfigsInputs.add(
    new Input()
        .withName("linkConfig.username")
        .withValue("username")
);
listConfigsInputs.add(
    new Input()
        .withName("linkConfig.password")
        .withValue("DB_password")
);
listConfigsInputs.add(
    new Input()
        .withName("linkConfig.fetchSize")
        .withValue("100000")
);
listConfigsInputs.add(
    new Input()
        .withName("linkConfig.usingNative")
        .withValue("false")
);
List<Configs> listLinkConfigValuesConfigs = new ArrayList<>();
listLinkConfigValuesConfigs.add(
    new Configs()
        .withInputs(listConfigsInputs)
        .withName("linkConfig")
);
LinksLinkconfigvalues linkConfigValuesLinks = new LinksLinkconfigvalues();
linkConfigValuesLinks.withConfigs(listLinkConfigValuesConfigs);
List<Links> listbodyLinks = new ArrayList<>();
listbodyLinks.add(
    new Links()
        .withLinkConfigValues(linkConfigValuesLinks)
        .withName("mysql_link")
        .withCreationDate(1496654788622L)
        .withConnectorName("generic-jdbc-connector")
        .withUpdateDate(1496654788622L)
        .withEnabled(true)
);
body.withLinks(listbodyLinks);
request.withBody(body);
try {
    CreateLinkResponse response = client.createLink(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

```
}  
}
```

## Python

创建一个名为mysql\_link的数据连接。

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcdm.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = CdmClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateLinkRequest()  
        request.cluster_id = "{cluster_id}"  
        listInputsConfigs = [  
            Input(  
                name="linkConfig.databaseType",  
                value="MYSQL"  
            ),  
            Input(  
                name="linkConfig.host",  
                value="100.94.8.163"  
            ),  
            Input(  
                name="linkConfig.port",  
                value="3306"  
            ),  
            Input(  
                name="linkConfig.database",  
                value="DB_name"  
            ),  
            Input(  
                name="linkConfig.username",  
                value="username"  
            ),  
            Input(  
                name="linkConfig.password",  
                value="DB_password"  
            ),  
            Input(  
                name="linkConfig.fetchSize",  
                value="100000"  
            ),  
            Input(  
                name="linkConfig.usingNative",  
                value="false"  
            )  
        ]  
    }
```

```
listConfigsLinkConfigValues = [
    Configs(
        inputs=listInputsConfigs,
        name="linkConfig"
    )
]
linkConfigValuesLinks = LinksLinkconfigvalues(
    configs=listConfigsLinkConfigValues
)
listLinksbody = [
    Links(
        link_config_values=linkConfigValuesLinks,
        name="mysql_link",
        creation_date=1496654788622,
        connector_name="generic-jdbc-connector",
        update_date=1496654788622,
        enabled=True
    )
]
request.body = CdmCreateAndUpdateLinkReq(
    links=listLinksbody
)
response = client.create_link(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

创建一个名为mysql\_link的数据连接。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateLinkRequest{}
    request.ClusterId = "{cluster_id}"
    valueInputs:= "MYSQL"
```

```

var valueInputsInterface interface{} = valueInputs
valueInputs1:= "100.94.8.163"
var valueInputsInterface1 interface{} = valueInputs1
valueInputs2:= "3306"
var valueInputsInterface2 interface{} = valueInputs2
valueInputs3:= "DB_name"
var valueInputsInterface3 interface{} = valueInputs3
valueInputs4:= "username"
var valueInputsInterface4 interface{} = valueInputs4
valueInputs5:= "DB_password"
var valueInputsInterface5 interface{} = valueInputs5
valueInputs6:= "100000"
var valueInputsInterface6 interface{} = valueInputs6
valueInputs7:= "false"
var valueInputsInterface7 interface{} = valueInputs7
var listInputsConfigs = []model.Input{
    {
        Name: "linkConfig.databaseType",
        Value: &valueInputsInterface,
    },
    {
        Name: "linkConfig.host",
        Value: &valueInputsInterface1,
    },
    {
        Name: "linkConfig.port",
        Value: &valueInputsInterface2,
    },
    {
        Name: "linkConfig.database",
        Value: &valueInputsInterface3,
    },
    {
        Name: "linkConfig.username",
        Value: &valueInputsInterface4,
    },
    {
        Name: "linkConfig.password",
        Value: &valueInputsInterface5,
    },
    {
        Name: "linkConfig.fetchSize",
        Value: &valueInputsInterface6,
    },
    {
        Name: "linkConfig.usingNative",
        Value: &valueInputsInterface7,
    },
}
var listConfigsLinkConfigValues = []model.Configs{
    {
        Inputs: listInputsConfigs,
        Name: "linkConfig",
    },
}
linkConfigValuesLinks := &model.LinksLinkconfigvalues{
    Configs: listConfigsLinkConfigValues,
}
creationDateLinks:= int64(1496654788622)
updateDateLinks:= int64(1496654788622)
enabledLinks:= true
var listLinksbody = []model.Links{
    {
        LinkConfigValues: linkConfigValuesLinks,
        Name: "mysql_link",
        CreationDate: &creationDateLinks,
        ConnectorName: "generic-jdbc-connector",
        UpdateDate: &updateDateLinks,
        Enabled: &enabledLinks,
    }
}

```

```
    },  
  }  
  request.Body = &model.CdmCreateAndUpdateLinkReq{  
    Links: listLinksbody,  
  }  
  response, err := client.CreateLink(request)  
  if err == nil {  
    fmt.Printf("%+v\n", response)  
  } else {  
    fmt.Println(err)  
  }  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.3.2 查询连接

#### 功能介绍

查询连接接口。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/link/{link\_name}

表 4-181 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
link_name	是	String	连接名称。

## 请求参数

表 4-182 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 200

表 4-183 响应 Body 参数

参数	参数类型	描述
links	Array of <a href="#">links</a> objects	连接列表，请参见links数据结构说明。
fromTo-unMapping	String	表/文件迁移不支持哪些数据源迁移到哪些数据源。
batchFromTo-mapping	String	整库迁移支持哪些数据源迁移到哪些数据源。

表 4-184 links

参数	参数类型	描述
link-config-values	<a href="#">link-config-values</a> object	连接参数配置，请参见link-config-values参数说明。
creation-user	String	创建连接的用户。
name	String	连接名称。



参数	参数类型	描述
id	Integer	连接ID。
creation-date	Long	创建连接的时间。
connector-name	String	连接器名称，对应的连接参数如下：generic-jdbc-connector：关系数据库连接。obs-connector：OBS连接。hdfs-connector：HDFS连接。hbase-connector：HBase连接、CloudTable连接。hive-connector：Hive连接。ftp-connector/sftp-connector：FTP/SFTP连接。mongodb-connector：MongoDB连接。redis-connector：Redis/DCS连接。kafka-connector：Kafka连接。dis-connector：DIS连接。elasticsearch-connector：Elasticsearch/云搜索服务连接。dli-connector：DLI连接。http-connector：HTTP/HTTPS连接，该连接暂无连接参数。dms-kafka-connector：DMSKafka连接。
update-date	Long	更新连接的时间。
enabled	Boolean	是否激活连接，默认为“true”。
update-user	String	更新连接的用户。

表 4-185 link-config-values

参数	参数类型	描述
configs	Array of <b>configs</b> objects	连接配置参数数据结构，请参见configs参数说明。
extended-configs	<b>extended-configs</b> object	扩展配置，请参见extended-configs参数说明。
validators	Array of strings	校验器。

表 4-186 configs

参数	参数类型	描述
inputs	Array of <b>Input</b> objects	输入参数列表，列表中的每个参数为“name,value”结构，请参考inputs数据结构参数说明。在“from-config-values”数据结构中，不同的源连接类型有不同的“inputs”参数列表，请参见源端作业参数说明下的章节。在“to-config-values”数据结构中，不同的目的连接类型有不同的“inputs”参数列表，请参见目的端作业参数说明下面的子章节。在“driver-config-values”数据结构中，“inputs”具体参数请参见作业任务参数说明。
name	String	配置名称：源端作业的配置名称为“fromJobConfig”。目的端作业的配置名称为“toJobConfig”，连接的配置名称固定为“linkConfig”。
id	Integer	配置ID，由系统生成，用户无需填写。
type	String	配置类型，由系统生成，用户无需填写。值为LINK或者JOB，如果是连接管理API，则为LINK；如果是作业管理API，则为JOB。

表 4-187 Input

参数	参数类型	描述
name	String	参数名： <ul style="list-style-type: none"> <li>如果是连接管理API，则以“linkConfig.”开头，对于不同连接类型有不同的参数，具体可参见<a href="#">连接参数说明</a>下相应连接的参数说明。</li> <li>如果是作业管理API，对于源端连接参数，则以“fromJobConfig.”开头，具体可参见<a href="#">源端作业参数说明</a>下相应的源端参数说明；对于目的端连接参数，则以“toJobConfig.”开头，具体可参见<a href="#">目的端作业参数说明</a>下相应的目的端参数说明；对于作业任务参数，请参见<a href="#">作业任务参数说明</a>下相应的任务参数说明。</li> </ul>
value	Object	参数值，参数名对应的值，必须填写为字符串。
type	String	值类型，如STRING、INTEGER，由系统设定，用户无需填写。

表 4-188 extended-configs

参数	参数类型	描述
name	String	名称。
value	String	值。

## 请求示例

```
GET /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/link/sftplink
```

## 响应示例

状态码： 200

OK。

```
{
  "links" : [ {
    "link-config-values" : {
      "configs" : [ {
        "inputs" : [ {
          "name" : "linkConfig.server",
          "type" : "STRING",
          "value" : "100.94.8.163"
        }, {
          "name" : "linkConfig.port",
          "type" : "INTEGER",
          "value" : 22
        }, {
          "name" : "linkConfig.username",
          "type" : "STRING",
          "value" : "root"
        }, {
          "name" : "linkConfig.password",
          "type" : "STRING",
          "value" : "Add password here"
        }
      ]
    },
    "name" : "linkConfig"
  } ]
},
"creation-user" : "cdm",
"name" : "sftp_link",
"creation-date" : 1516674482640,
"connector-name" : "sftp-connector",
"update-date" : 1516674476022,
"enabled" : true,
"update-user" : "cdm"
} ] }
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class ShowLinkSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowLinkRequest request = new ShowLinkRequest();
        request.withClusterId("{cluster_id}");
        request.withLinkName("{link_name}");
        try {
            ShowLinkResponse response = client.showLink(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = CdmClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CdmRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowLinkRequest()
    request.cluster_id = "{cluster_id}"
    request.link_name = "{link_name}"
    response = client.show_link(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowLinkRequest{}
    request.ClusterId = "{cluster_id}"
    request.LinkName = "{link_name}"
    response, err := client.ShowLink(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.3.3 删除连接

#### 功能介绍

删除连接接口。

#### 调用方法

请参见[如何调用API](#)。

#### URI

DELETE /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/link/{link\_name}

表 4-189 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
link_name	是	String	需要删除的连接名。

## 请求参数

表 4-190 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

## 响应参数

状态码： 500

表 4-191 响应 Body 参数

参数	参数类型	描述
errCode	String	错误码。
externalMessage	String	错误描述。

## 请求示例

```
DELETE /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/link/jdbclink
```

## 响应示例

状态码： 500

服务内部错误，具体返回错误码请参考错误码。

```
{
  "errCode": "Cdm.0021",
  "externalMessage": "Given link name is in use"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
```

```
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

public class DeleteLinkSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteLinkRequest request = new DeleteLinkRequest();
        request.withClusterId("{cluster_id}");
        request.withLinkName("{link_name}");
        try {
            DeleteLinkResponse response = client.deleteLink(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
```



```
.with_region(CdmRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = DeleteLinkRequest()
    request.cluster_id = "{cluster_id}"
    request.link_name = "{link_name}"
    response = client.delete_link(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cdm.NewCdmClient(
        cdm.CdmClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteLinkRequest{}
    request.ClusterId = "{cluster_id}"
    request.LinkName = "{link_name}"
    response, err := client.DeleteLink(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

### 4.3.4 修改连接

#### 功能介绍

修改连接接口。

#### 调用方法

请参见[如何调用API](#)。

#### URI

PUT /v1.1/{project\_id}/clusters/{cluster\_id}/cdm/link/{link\_name}

表 4-192 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
cluster_id	是	String	集群ID。
link_name	是	String	连接名称。

## 请求参数

表 4-193 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-194 请求 Body 参数

参数	是否必选	参数类型	描述
links	是	Array of <a href="#">links</a> objects	连接列表，请参见links数据结构说明。

表 4-195 links

参数	是否必选	参数类型	描述
link-config-values	是	<a href="#">link-config-values</a> object	连接参数配置，请参见link-config-values参数说明。
creation-user	否	String	创建连接的用户。
name	是	String	连接名称。
id	否	Integer	连接ID。
creation-date	否	Long	创建连接的时间。

参数	是否必选	参数类型	描述
connector-name	是	String	连接器名称，对应的连接参数如下： generic-jdbc-connector: 关系数据库连接。 obs-connector: OBS连接。 hdfs-connector: HDFS连接。 hbase-connector: HBase连接、CloudTable连接。 hive-connector: Hive连接。 ftp-connector/sftp-connector: FTP/SFTP连接。 mongodb-connector: MongoDB连接。 redis-connector: Redis/DCS连接。 kafka-connector: Kafka连接。 dis-connector: DIS连接。 elasticsearch-connector: Elasticsearch/云搜索服务连接。 dli-connector: DLI连接。 http-connector: HTTP/HTTPS连接，该连接暂无连接参数。 dms-kafka-connector: DMSKafka连接。
update-date	否	Long	更新连接的时间。
enabled	否	Boolean	是否激活连接，默认为“true”。
update-user	否	String	更新连接的用户。

表 4-196 link-config-values

参数	是否必选	参数类型	描述
configs	是	Array of <b>configs</b> objects	连接配置参数数据结构，请参见configs参数说明。
extended-configs	否	<b>extended-configs</b> object	扩展配置，请参见extended-configs参数说明。
validators	否	Array of strings	校验器。

表 4-197 configs

参数	是否必选	参数类型	描述
inputs	是	Array of <b>Input</b> objects	输入参数列表，列表中的每个参数为“name,value”结构，请参考inputs数据结构参数说明。在“from-config-values”数据结构中，不同的源连接类型有不同的“inputs”参数列表，请参见源端作业参数说明下的章节。在“to-config-values”数据结构中，不同的目的连接类型有不同的“inputs”参数列表，请参见目的端作业参数说明下面的子章节。在“driver-config-values”数据结构中，“inputs”具体参数请参见作业任务参数说明。
name	是	String	配置名称：源端作业的配置名称为“fromJobConfig”。目的端作业的配置名称为“toJobConfig”，连接的配置名称固定为“linkConfig”。
id	否	Integer	配置ID，由系统生成，用户无需填写。
type	否	String	配置类型，由系统生成，用户无需填写。值为LINK或者JOB，如果是连接管理API，则为LINK；如果是作业管理API，则为JOB。

表 4-198 Input

参数	是否必选	参数类型	描述
name	是	String	<p>参数名:</p> <ul style="list-style-type: none"> <li>如果是连接管理API, 则以“linkConfig.”开头, 对于不同连接类型有不同的参数, 具体可参见<a href="#">连接参数说明</a>下相应连接的参数说明。</li> <li>如果是作业管理API, 对于源端连接参数, 则以“fromJobConfig.”开头, 具体可参见<a href="#">源端作业参数说明</a>下相应的源端参数说明; 对于目的端连接参数, 则以“toJobConfig.”开头, 具体可参见<a href="#">目的端作业参数说明</a>下相应的目的端参数说明; 对于作业任务参数, 请参见<a href="#">作业任务参数说明</a>下相应的任务参数说明。</li> </ul>
value	是	Object	参数值, 参数名对应的值, 必须填写为字符串。
type	否	String	值类型, 如STRING、INTEGER, 由系统设定, 用户无需填写。

表 4-199 extended-configs

参数	是否必选	参数类型	描述
name	否	String	名称。
value	否	String	值。

## 响应参数

状态码: 200

表 4-200 响应 Body 参数

参数	参数类型	描述
validation-result	Array of <a href="#">validationResult</a> objects	校验结构: 如果创建连接失败, 返回失败原因, 请参见validation-result参数说明。如果创建成功, 返回空列表。

表 4-201 validationResult

参数	参数类型	描述
linkConfig	Array of <a href="#">validationLinkConfig</a> objects	创建或更新连接校验结果，请参见linkConfig参数说明。

表 4-202 validationResultLinkConfig

参数	参数类型	描述
message	String	错误描述。
status	String	错误级别，如：ERROR、WARNING。

状态码： 500

表 4-203 响应 Body 参数

参数	参数类型	描述
message	String	错误描述。
status	String	错误级别，如：ERROR、WARNING。

## 请求示例

修改一个名为mysql\_link的数据连接。

```
PUT /v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/link/rdb_link
```

```
{
  "links": [ {
    "link-config-values": {
      "configs": [ {
        "inputs": [ {
          "name": "linkConfig.databaseType",
          "value": "MYSQL"
        }, {
          "name": "linkConfig.host",
          "value": "100.94.8.163"
        }, {
          "name": "linkConfig.port",
          "value": "3306"
        }, {
          "name": "linkConfig.database",
          "value": "DB_name"
        }, {
          "name": "linkConfig.username",
          "value": "username"
        }, {
          "name": "linkConfig.password",
```

```

        "value" : "DB_password"
      }, {
        "name" : "linkConfig.fetchSize",
        "value" : "100000"
      }, {
        "name" : "linkConfig.usingNative",
        "value" : "false"
      }
    ],
    "name" : "linkConfig"
  }
}
},
"name" : "mysql_link",
"creation-date" : 1496654788622,
"connector-name" : "generic-jdbc-connector",
"update-date" : 1496654788622,
"enabled" : true
}]
}

```

## 响应示例

**状态码： 200**

OK。

```

{
  "validation-result" : [ {} ]
}

```

**状态码： 500**

服务内部错误，具体返回错误码请参考错误码。

```

{
  "validation-result" : [ {
    "linkConfig" : [ {
      "message" : "Can't connect to the database with given credentials: The authentication type 12 is not supported. Check that you have configured the pg_hba.conf file to include the client's IP address or subnet, and that it is using an authentication scheme supported by the driver.",
      "status" : "ERROR"
    }
  ]
}
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

修改一个名为mysql\_link的数据连接。

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cdm.v1.region.CdmRegion;
import com.huaweicloud.sdk.cdm.v1.*;
import com.huaweicloud.sdk.cdm.v1.model.*;

import java.util.List;
import java.util.ArrayList;

```



```
public class UpdateLinkSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CdmClient client = CdmClient.newBuilder()
            .withCredential(auth)
            .withRegion(CdmRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateLinkRequest request = new UpdateLinkRequest();
        request.withClusterId("{cluster_id}");
        request.withLinkName("{link_name}");
        CdmCreateAndUpdateLinkReq body = new CdmCreateAndUpdateLinkReq();
        List<Input> listConfigsInputs = new ArrayList<>();
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.databaseType")
                .withValue("MYSQL")
        );
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.host")
                .withValue("100.94.8.163")
        );
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.port")
                .withValue("3306")
        );
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.database")
                .withValue("DB_name")
        );
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.username")
                .withValue("username")
        );
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.password")
                .withValue("DB_password")
        );
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.fetchSize")
                .withValue("100000")
        );
        listConfigsInputs.add(
            new Input()
                .withName("linkConfig.usingNative")
                .withValue("false")
        );
        List<Configs> listLinkConfigValuesConfigs = new ArrayList<>();
        listLinkConfigValuesConfigs.add(
            new Configs()
        );
    }
}
```

```
        .withInputs(listConfigsInputs)
        .withName("linkConfig")
    );
    LinksLinkconfigvalues linkConfigValuesLinks = new LinksLinkconfigvalues();
    linkConfigValuesLinks.withConfigs(listLinkConfigValuesConfigs);
    List<Links> listbodyLinks = new ArrayList<>();
    listbodyLinks.add(
        new Links()
            .withLinkConfigValues(linkConfigValuesLinks)
            .withName("mysql_link")
            .withCreationDate(1496654788622L)
            .withConnectorName("generic-jdbc-connector")
            .withUpdateDate(1496654788622L)
            .withEnabled(true)
    );
    body.withLinks(listbodyLinks);
    request.withBody(body);
    try {
        UpdateLinkResponse response = client.updateLink(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

修改一个名为mysql\_link的数据连接。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcdm.v1.region.cdm_region import CdmRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcdm.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = CdmClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CdmRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateLinkRequest()
        request.cluster_id = "{cluster_id}"
        request.link_name = "{link_name}"
        listInputsConfigs = [
            Input(
```

```
        name="linkConfig.databaseType",
        value="MYSQL"
    ),
    Input(
        name="linkConfig.host",
        value="100.94.8.163"
    ),
    Input(
        name="linkConfig.port",
        value="3306"
    ),
    Input(
        name="linkConfig.database",
        value="DB_name"
    ),
    Input(
        name="linkConfig.username",
        value="username"
    ),
    Input(
        name="linkConfig.password",
        value="DB_password"
    ),
    Input(
        name="linkConfig.fetchSize",
        value="100000"
    ),
    Input(
        name="linkConfig.usingNative",
        value="false"
    )
]
listConfigsLinkConfigValues = [
    Configs(
        inputs=listInputsConfigs,
        name="linkConfig"
    )
]
linkConfigValuesLinks = LinksLinkconfigvalues(
    configs=listConfigsLinkConfigValues
)
listLinksbody = [
    Links(
        link_config_values=linkConfigValuesLinks,
        name="mysql_link",
        creation_date=1496654788622,
        connector_name="generic-jdbc-connector",
        update_date=1496654788622,
        enabled=True
    )
]
request.body = CdmCreateAndUpdateLinkReq(
    links=listLinksbody
)
response = client.update_link(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

修改一个名为mysql\_link的数据连接。

```
package main

import (
```

```
"fmt"  
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
cdm "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1"  
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/model"  
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cdm/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := cdm.NewCdmClient(  
        cdm.CdmClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build()  
    )  
  
    request := &model.UpdateLinkRequest{}  
    request.ClusterId = "{cluster_id}"  
    request.LinkName = "{link_name}"  
    valueInputs := "MYSQL"  
    var valueInputsInterface interface{} = valueInputs  
    valueInputs1 := "100.94.8.163"  
    var valueInputsInterface1 interface{} = valueInputs1  
    valueInputs2 := "3306"  
    var valueInputsInterface2 interface{} = valueInputs2  
    valueInputs3 := "DB_name"  
    var valueInputsInterface3 interface{} = valueInputs3  
    valueInputs4 := "username"  
    var valueInputsInterface4 interface{} = valueInputs4  
    valueInputs5 := "DB_password"  
    var valueInputsInterface5 interface{} = valueInputs5  
    valueInputs6 := "100000"  
    var valueInputsInterface6 interface{} = valueInputs6  
    valueInputs7 := "false"  
    var valueInputsInterface7 interface{} = valueInputs7  
    var listInputsConfigs = []model.Input{  
        {  
            Name: "linkConfig.databaseType",  
            Value: &valueInputsInterface,  
        },  
        {  
            Name: "linkConfig.host",  
            Value: &valueInputsInterface1,  
        },  
        {  
            Name: "linkConfig.port",  
            Value: &valueInputsInterface2,  
        },  
        {  
            Name: "linkConfig.database",  
            Value: &valueInputsInterface3,  
        },  
        {  
            Name: "linkConfig.username",  
            Value: &valueInputsInterface4,  
        },  
    },  
}
```

```

    {
        Name: "linkConfig.password",
        Value: &valueInputsInterface5,
    },
    {
        Name: "linkConfig.fetchSize",
        Value: &valueInputsInterface6,
    },
    {
        Name: "linkConfig.usingNative",
        Value: &valueInputsInterface7,
    },
}
var listConfigsLinkConfigValues = []model.Configs{
    {
        Inputs: listInputsConfigs,
        Name: "linkConfig",
    },
}
linkConfigValuesLinks := &model.LinksLinkconfigvalues{
    Configs: listConfigsLinkConfigValues,
}
creationDateLinks:= int64(1496654788622)
updateDateLinks:= int64(1496654788622)
enabledLinks:= true
var listLinksbody = []model.Links{
    {
        LinkConfigValues: linkConfigValuesLinks,
        Name: "mysql_link",
        CreationDate: &creationDateLinks,
        ConnectorName: "generic-jdbc-connector",
        UpdateDate: &updateDateLinks,
        Enabled: &enabledLinks,
    },
}
request.Body = &model.CdmCreateAndUpdateLinkReq{
    Links: listLinksbody,
}
response, err := client.UpdateLink(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。

状态码	描述
500	服务内部错误，具体返回错误码请参考错误码。
503	服务不可用。

## 错误码

请参见[错误码](#)。

## 4.4 附：公共数据结构

### 4.4.1 连接参数说明

#### 4.4.1.1 关系数据库连接

##### 介绍

通过JDBC连接，可以对以下关系型数据库抽取、加载数据：

- 云数据库 PostgreSQL
- 云数据库 SQL Server
- PostgreSQL
- Microsoft SQL Server

##### 连接样例

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.databaseType",
                "value": "MYSQL"
              },
              {
                "name": "linkConfig.host",
                "value": "10.120.205.30"
              },
              {
                "name": "linkConfig.port",
                "value": "3306"
              },
              {
                "name": "linkConfig.database",
                "value": "DB_name"
              },
              {
                "name": "linkConfig.username",
                "value": "username"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```

    {
      "name": "linkConfig.password",
      "value": "Add password here"
    },
    {
      "name": "linkConfig.fetchSize",
      "value": "100000"
    },
    {
      "name": "linkConfig.commitSize",
      "value": "10000"
    },
    {
      "name": "linkConfig.usingNative",
      "value": "false"
    },
    {
      "name": "linkConfig.useSSL",
      "value": "false"
    }
  ],
  "name": "linkConfig"
}
],
"name": "mysql_link",
"connector-name": "generic-jdbc-connector"
}
]
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.data baseType	是	枚举	数据库类型： <ul style="list-style-type: none"> <li>• ORACLE</li> <li>• MYSQL</li> <li>• SQLSERVER</li> <li>• DB2</li> <li>• POSTGRESQL</li> <li>• DWS</li> <li>• DDM</li> <li>• SAP HANA</li> </ul>
linkConfig.host	是	String	数据库服务器地址。
linkConfig.port	是	String	数据库服务器的端口号。
linkConfig.data baseconfig	否	枚举	创建Oracle连接时才有该参数，选择Oracle数据库连接类型： <ul style="list-style-type: none"> <li>• SERVICENAME：通过 SERVICE_NAME连接Oracle数据库。</li> <li>• SID：通过SID连接Oracle数据库。</li> </ul>

参数	是否必选	类型	说明
linkConfig.sidname	否	String	配置Oracle实例ID，用于实例区分各个数据库。创建Oracle连接，且linkConfig.databaseconfig（数据库连接类型）选择为“SID”时才有该参数。
linkConfig.database	否	String	数据库名称。
linkConfig.username	是	String	用户名。
linkConfig.password	是	String	用户密码。
linkConfig.fetchSize	否	String	每次请求获取的数据行数。
linkConfig.commitSize	否	String	每次请求提交的数据行数。
linkConfig.usingNative	否	Boolean	<p>是否使用数据库本地API加速。</p> <p>创建MySQL连接时，使用本地API加速，可以使用MySQL的LOAD DATA功能加快数据导入，提高导入数据到MySQL数据库的性能。</p> <p><b>说明</b></p> <p>REPLACE 和 IGNORE 修饰符用于处理与现有行具有相同唯一键值（PRIMARY KEY或UNIQUE索引值）的新输入的行。</p> <ul style="list-style-type: none"> <li>• 约束冲突处理配置为"replace into"或"local"，使用REPLACE，与现有行中的唯一键值具有相同值的新行将替换现有行。</li> <li>• 约束冲突处理配置为"insert into"，默认使用IGNORE，与唯一键值上的现有行重复的新行将被丢弃，任务不会终止。因为Mysql服务机制无法在操作过程中停止文件传输，此情况下，CDM界面显示写入记录与实际更新行数会不一致。</li> </ul> <p>Mysql Local模式详情请参见： <a href="https://dev.mysql.com/doc/refman/8.0/en/load-data.html">https://dev.mysql.com/doc/refman/8.0/en/load-data.html</a></p>
linkConfig.isRds	否	Boolean	是否支持RDS服务，云数据库默认true，其余默认false



参数	是否必选	类型	说明
linkConfig.useSSL	否	Boolean	是否使用加密传输，支持对RDS服务启用SSL加密传输，仅在创建dws连接时才有此参数。
linkConfig.jdbcProperties	否	Map	连接属性，指定数据源的JDBC连接器的属性，参考对应数据库的JDBC连接器说明文档进行配置。
linkConfig.version	否	枚举	<p>创建Oracle连接时才有该参数，根据您Oracle数据库的版本来选择。</p> <ul style="list-style-type: none"> <li>• HIGH_VERSION: 当您的Oracle数据库版本高于12.1时，选择该值。</li> <li>• MED_VERSION: 当您的Oracle数据库版本为12.1时，选择该值。</li> <li>• LOW_VERSION: 当您的Oracle数据库版本低于12.1时，选择该值。</li> </ul> <p>当出现“java.sql.SQLException: Protocol violation异常”时，可以尝试更换版本号。</p>
dialect.identifierEnclose	否	String	引用符号，连接引用表名或列名时的分隔符号，参考对应数据库的产品文档进行配置。

#### 4.4.1.2 OBS 连接

##### 介绍

通过OBS连接，可以对对象存储服务（Object Storage Service，简称OBS）抽取或加载文件，支持CSV、JSON和二进制格式。

##### 连接样例

本示例为连接样例消息体。在实际使用中，AK（accessKey）和SK（securityKey）建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.storageType",
                "value": "OBS"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```

    {
      "name": "linkConfig.obsBucketType",
      "value": "PFS"
    },
    {
      "name": "linkConfig.server",
      "value": "10.121.16.183"
    },
    {
      "name": "linkConfig.port",
      "value": "443"
    },
    {
      "name": "linkConfig.accessKey",
      "value": "<YOUR AK>"
    },
    {
      "name": "linkConfig.securityKey",
      "value": "<YOUR SK>"
    }
  ],
  "name": "linkConfig"
}
],
"name": "obs_link",
"connector-name": "obs-connector"
}
]
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.storageType	是	String	对象存储的类型。
linkConfig.obsBucketType	否	String	OBS桶类型。 <ul style="list-style-type: none"> <li>对象桶：OB。</li> <li>并行文件系统：PFS。</li> </ul> <b>说明</b> 桶类型分为对象桶和并行文件系统，对象桶过滤文件会比较慢，建议客户使用并行文件桶。
linkConfig.server	是	String	OBS服务器的终端节点（Endpoint）。
linkConfig.port	是	String	数据传输协议端口，https是443，http是80。
linkConfig.accessKey	是	String	访问标识（AK）。建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
linkConfig.securityKey	是	String	密钥（SK）。建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。

### 4.4.1.3 HDFS 连接

#### 介绍

通过HDFS连接，可以对MRS、FusionInsight HD或开源Hadoop的HDFS抽取、加载文件，支持CSV、Parquet和二进制格式。

#### 连接样例

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.hadoopType",
                "value": "FusionInsight HD"
              },
              {
                "name": "linkConfig.host",
                "value": "10.120.205.143"
              },
              {
                "name": "linkConfig.casPort",
                "value": "20009"
              },
              {
                "name": "linkConfig.port",
                "value": "28443"
              },
              {
                "name": "linkConfig.authType",
                "value": "KERBEROS"
              },
              {
                "name": "linkConfig.user",
                "value": "admin"
              },
              {
                "name": "linkConfig.password",
                "value": "Add password here"
              },
              {
                "name": "linkConfig.runMode",
                "value": "STANDALONE"
              }
            ],
            "name": "linkConfig"
          }
        ]
      },
      "name": "hdfslink",
      "connector-name": "hdfs-connector"
    }
  ]
}
```

## 连接参数

参数	是否必选	类型	说明
linkConfig.hadoopType	是	枚举	Hadoop类型： <ul style="list-style-type: none"> <li>• MRS：表示连接MRS的HDFS。</li> <li>• FusionInsight HD：表示连接FusionInsight HD的HDFS。</li> <li>• Apache Hadoop：表示连接开源Apache Hadoop的HDFS。</li> </ul>
linkConfig.uri	否	String	连接Apache Hadoop时的Namenode URI地址，格式为“ip:port”。
linkConfig.host	否	String	连接MRS或FusionInsight HD时，需要配置Manager平台的IP地址。
linkConfig.port	否	String	连接FusionInsight HD时，需要配置Manager平台的端口。
linkConfig.casPort	否	String	连接FusionInsight HD时，需要配置与FusionInsight HD对接的CAS Server的端口。
linkConfig.user	否	String	登录Manager平台的用户名，使用集群配置时不用配置
linkConfig.password	否	String	登录Manager平台的密码，使用集群配置时不用配置
linkConfig.authType	否	枚举	认证类型，分为以下两种： <ul style="list-style-type: none"> <li>• Simple：非安全模式选择Simple鉴权。</li> <li>• Kerberos：安全模式选择Kerberos鉴权。</li> </ul>
linkConfig.principal	否	String	Kerberos认证所需的Principal，您也可以联系管理员获取此账号。使用集群配置前需在集群配置管理中配置此参数。
linkConfig.keytab	否	FileContent	Kerberos认证所需的keytab文件的本地绝对路径，您也可以联系管理员获取此文件。使用集群配置前需在集群配置管理中配置此参数。

参数	是否必选	类型	说明
linkConfig.run Mode	否	枚举	<p>选择HDFS连接的运行模式：</p> <ul style="list-style-type: none"> <li>EMBEDDED：连接实例与CDM运行在一起，该模式性能较好。</li> <li>STANDALONE：连接实例运行在独立进程。如果CDM需要对接多个Hadoop数据源（MRS、Hadoop或CloudTable），并且既有KERBEROS认证模式又有SIMPLE认证模式，只能使用STANDALONE模式。选择STANDALONE模式时，CDM支持在多个MRS集群的HDFS之间迁移数据。</li> </ul> <p>若在一个CDM中同时连接两个及以上开启Kerberos认证且realm相同的集群，只能使用EMBEDDED运行模式连接其中一个集群，其余需使用STANDALONE。</p>
linkConfig.properties	否	Map	属性配置，可以添加客户端的配置属性，所添加的每个属性需配置属性名称和值

#### 4.4.1.4 HBase 连接

##### 介绍

通过HBase连接，可以对MRS、FusionInsight HD、Apache Hadoop的HBase抽取、加载数据。

##### 连接样例

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.hbaseType",
                "value": "MRS"
              },
              {
                "name": "linkConfig.host",
                "value": "192.168.0.34"
              },
              {
                "name": "linkConfig.user",

```

```

        "value": "zephyr"
    },
    {
        "name": "linkConfig.password",
        "value": "Add password here."
    },
    {
        "name": "linkConfig.authType",
        "value": "KERBEROS"
    },
    {
        "name": "linkConfig.serviceType",
        "value": "HDFS"
    },
    {
        "name": "linkConfig.hBaseVersion",
        "value": "HBASE_2_X"
    },
    {
        "name": "linkConfig.runMode",
        "value": "EMBEDDED"
    }
],
"name": "linkConfig"
}
],
"extended-configs": {
    "name": "linkConfig.extendedFields",
    "value":
"eyJL1c2VDhHVzdGVyQ29uZmlnIjoiZmFsc2UiIjCLjBHVzdGVyQ29uZmlnUHlpbmNpcGFsIjoiemVwaHlyIn0="
}
},
"name": "mrs_hbase_dlf",
"connector-name": "hbase-connector"
}
]
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.hbaseType	是	枚举	HBase类型： <ul style="list-style-type: none"> <li>CloudTable：表示连接CloudTable服务。</li> <li>MRS：表示连接MRS的HBase。</li> <li>FusionInsight HD：表示连接FusionInsight HD的HBase。</li> <li>Apache Hadoop：表示连接开源Apache Hadoop的HBase。</li> </ul>
linkConfig.uri	否	String	连接Apache Hadoop时的Namenode URI地址，格式为“ip:port”。
linkConfig.host	否	String	连接MRS或FusionInsight HD时，需要配置Manager平台的IP地址。
linkConfig.port	否	String	连接FusionInsight HD时，需要配置Manager平台的端口。

参数	是否必选	类型	说明
linkConfig.casPort	否	String	连接FusionInsight HD时，需要配置与FusionInsight HD对接的CAS Server的端口。
linkConfig.hBaseVersion	是	枚举	HBase版本： <ul style="list-style-type: none"> <li>• HBASE_1_X</li> <li>• HBASE_2_X</li> </ul>
linkConfig.userName	否	String	登录Manager平台的用户名，使用集群配置时不用配置
linkConfig.password	否	String	登录Manager平台的密码，使用集群配置时不用配置
linkConfig.authType	否	枚举	认证类型，分为以下两种： <ul style="list-style-type: none"> <li>• Simple: 非安全模式选择Simple鉴权。</li> <li>• Kerberos: 安全模式选择Kerberos鉴权。</li> </ul>
linkConfig.principal	否	String	Kerberos认证所需的Principal，您也可以联系管理员获取此账号。
linkConfig.keytab	否	FileContent	Kerberos认证所需的keytab文件的本地绝对路径，您也可以联系管理员获取此文件。
linkConfig.serviceType	否	String	服务类型。目前支持HDFS和HBase。

参数	是否必选	类型	说明
linkConfig.run Mode	否	枚举	<p>“HBase_2_X”版本支持该参数。支持以下模式：</p> <ul style="list-style-type: none"> <li>EMBEDDED：连接实例与CDM运行在一起，该模式性能较好。</li> <li>STANDALONE：连接实例运行在独立进程。如果CDM需要对接多个Hadoop数据源（MRS、Hadoop或CloudTable），并且既有KERBEROS认证模式又有SIMPLE认证模式，只能使用STANDALONE模式。选择STANDALONE模式时，CDM支持在多个MRS集群的HDFS之间迁移数据。</li> </ul> <p>若在一个CDM中同时连接两个及以上开启Kerberos认证且realm相同的集群，只能使用EMBEDDED运行模式连接其中一个集群，其余需使用STANDALONE。</p>
linkConfig.properties	否	Map	属性配置，可以添加客户端的配置属性，所添加的每个属性需配置属性名称和值

#### 4.4.1.5 CloudTable 连接

##### 介绍

通过CloudTable连接，可以对CloudTable服务抽取、加载数据。

##### 连接样例

本示例为连接样例消息体。在实际使用中，AK（accessKey）和SK（securityKey）建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.hbaseType",
                "value": "CloudTable"
              },
              {
                "name": "linkConfig.zookeeperQuorum",
                "value": "cloudtable-pass-zk2-bae54VGN.cloudtable.com:2181,cloudtable-pass-zk1-Fu828so2.cloudtable.com:2181"
              }
            ]
          }
        ]
      }
    }
  ]
}
```



```

    },
    {
      "name": "linkConfig.iamAuth",
      "value": "true"
    },
    {
      "name": "linkConfig.cloudtableUser",
      "value": "zane"
    },
    {
      "name": "linkConfig.accessKey",
      "value": "<YOUR AK>"
    },
    {
      "name": "linkConfig.securityKey",
      "value": "<YOUR SK>"
    },
    {
      "name": "linkConfig.runMode",
      "value": "EMBEDDED"
    }
  ],
  "name": "linkConfig"
}
]
},
"name": "cloudtablelink",
"connector-name": "hbase-connector"
}
}
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.hbaseType	是	枚举	HBase类型： <ul style="list-style-type: none"> <li>• CloudTable：表示连接CloudTable服务。</li> <li>• MRS：表示连接MRS。</li> <li>• FusionInsight HD：表示连接FusionInsight HD。</li> <li>• Apache Hadoop：表示连接开源Apache Hadoop。</li> </ul>
linkConfig.zookeeperQuorum	是	String	连接“CloudTable”时必选，表示CloudTable的ZooKeeper链接地址。
linkConfig.iamAuth	是	Boolean	当选择IAM统一身份认证时，需要输入用户名、AK和SK。

参数	是否必选	类型	说明
linkConfig.runMode	是	枚举	选择HBase连接的运行模式： <ul style="list-style-type: none"> <li>EMBEDDED：连接实例与CDM运行在一起，该模式性能较好。</li> <li>STANDALONE：连接实例运行在独立进程。如果CDM需要对接多个Hadoop数据源（MRS、Hadoop或CloudTable），并且既有KERBEROS认证模式又有SIMPLE认证模式，只能使用STANDALONE模式。</li> </ul>
linkConfig.cloudtableUser	是	String	登录CloudTable集群的用户名。
linkConfig.accessKey	是	String	登录CloudTable集群的访问标识。建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
linkConfig.securityKey	是	String	登录CloudTable集群的密钥。建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。

#### 4.4.1.6 Hive 连接

##### 介绍

通过Hive连接，可以对MRS的Hive数据源进行抽取、加载数据。

##### 连接样例

本示例为连接样例消息体。在实际使用中，AK（accessKey）和SK（securityKey）建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.host",
                "value": "10.120.205.230"
              },
              {
                "name": "linkConfig.authType",
                "value": "KERBEROS"
              },
              {
                "name": "linkConfig.user",
                "value": "cdm"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```

        "name": "linkConfig.password",
        "value": "Add password here"
    }
    ],
    "name": "linkConfig"
}
]
},
"name": "hive_link",
"connector-name": "hive-connector"
}
]
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.host	是	String	MRS Manager的IP地址。
linkConfig.authType	是	枚举	访问MRS的认证类型： <ul style="list-style-type: none"> <li>• SIMPLE：非安全模式选择Simple鉴权。</li> <li>• KERBEROS：安全模式选择Kerberos鉴权。</li> </ul>
linkConfig.principal	否	String	Kerberos认证所需的Principal，您也可以联系管理员获取此账号。使用集群配置前需在集群配置管理中配置此参数。
linkConfig.keytab	否	FileContent	Kerberos认证所需的keytab文件的本地绝对路径，您也可以联系管理员获取此文件。使用集群配置前需在集群配置管理中配置此参数。
linkConfig.hiveVersion	是	枚举	Hive版本： <ul style="list-style-type: none"> <li>• HIVE_1_X</li> <li>• HIVE_3_X</li> </ul>
linkConfig.user	否	String	登录Manager平台的用户名，使用集群配置时不用配置
linkConfig.password	否	String	登录Manager平台的密码，使用集群配置时不用配置
linkConfig.uri	否	String	连接Apache Hadoop时的Namenode URI地址，格式为“ip:port”
linkConfig.hiveMsUris	否	String	连接Apache Hadoop时的Hive元数据地址，参考hive.metastore.uris配置项。例如：thrift://host-192-168-1-212:9083

参数	是否必选	类型	说明
linkConfig.obsSupport	是	Boolean	需服务端支持OBS存储。在创建Hive表时，您可以指定将表存储在OBS中。
linkConfig.runMode	是	枚举	<p>“HIVE_3_X”版本支持该参数。支持以下模式：</p> <ul style="list-style-type: none"> <li>EMBEDDED：连接实例与CDM运行在一起，该模式性能较好。</li> <li>STANDALONE：连接实例运行在独立进程。如果CDM需要对接多个Hadoop数据源（MRS、Hadoop或CloudTable），并且既有KERBEROS认证模式又有SIMPLE认证模式，只能使用STANDALONE模式。</li> </ul> <p>说明：STANDALONE模式主要是用来解决版本冲突问题的运行模式。当同一种数据连接的源端或者目的端连接器的版本不一致时，存在jar包冲突的情况，这时需要将源端或目的端放在STANDALONE进程里，防止冲突导致迁移失败。</p>
linkConfig.accessKey	否	String	访问标识（AK）。obs支持选是时需配置此参数。建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
linkConfig.securityKey	否	String	密钥（SK）。obs支持选是时需配置此参数。建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
linkConfig.properties	否	Map	属性配置，可以添加客户端的配置属性，所添加的每个属性需配置属性名称和值

#### 4.4.1.7 FTP/SFTP 连接

##### 介绍

通过FTP或SFTP连接，可以对FTP或SFTP服务器抽取或加载文件，支持CSV、二进制和JSON格式。

##### 连接样例

```
{
  "links": [
```

```

{
  "link-config-values": {
    "configs": [
      {
        "inputs": [
          {
            "name": "linkConfig.server",
            "value": "10.120.85.167"
          },
          {
            "name": "linkConfig.port",
            "value": "22"
          },
          {
            "name": "linkConfig.username",
            "value": "username"
          },
          {
            "name": "linkConfig.password",
            "value": "Add password here"
          }
        ],
        "name": "linkConfig"
      }
    ],
    "name": "sftp_link",
    "connector-name": "sftp-connector"
  }
}

```

## 连接参数

FTP和SFTP的连接参数相同。

参数	是否必选	类型	说明
linkConfig.server	是	String	FTP或SFTP服务器的地址。
linkConfig.port	是	String	FTP或SFTP服务器端口号。
linkConfig.userName	是	String	登录FTP或SFTP服务器的用户名。
linkConfig.password	是	String	登录用户的密码。

### 4.4.1.8 MongoDB 连接

#### 介绍

通过MongoDB连接，可以对MongoDB服务器抽取、加载数据。

#### 连接样例

```

{
  "links": [
    {
      "link-config-values": {
        "configs": [

```

```

    {
      "inputs": [
        {
          "name": "linkConfig.serverList",
          "value": "10.120.84.149:27017"
        },
        {
          "name": "linkConfig.database",
          "value": "DB_name"
        },
        {
          "name": "linkConfig.userName",
          "value": "username"
        },
        {
          "name": "linkConfig.password",
          "value": "Add password here"
        }
      ],
      "name": "linkConfig"
    }
  ],
  "name": "mongo_link",
  "connector-name": "mongodb-connector"
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.serverList	是	String	服务器地址列表，格式如：“host1:port1;host2:port2”。
linkConfig.database	是	String	MongoDB的数据库名称。
linkConfig.userName	是	String	连接MongoDB服务器的用户名。
linkConfig.password	是	String	连接MongoDB服务器的密码。

### 4.4.1.9 Redis 连接

#### 介绍

通过Redis连接，可以对Redis服务器抽取或加载数据。

#### 连接样例

```

{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {

```

```

        "name": "linkConfig.deploymentMode",
        "value": "Cluster"
    },
    {
        "name": "linkConfig.serverlist",
        "value": "10.120.84.149:7300"
    },
    {
        "name": "linkConfig.password",
        "value": "Add password here"
    },
    {
        "name": "linkConfig.dbIndex",
        "value": "0"
    }
  ],
  "name": "linkConfig"
}
]
},
"name": "redis_link",
"connector-name": "redis-connector"
}
]
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.deploymentMode	是	枚举	Redis部署方式： <ul style="list-style-type: none"> <li>• Single：表示单机部署。</li> <li>• Cluster：表示集群部署。</li> </ul>
linkConfig.serverlist	是	String	服务器地址列表，格式如：“host1:port1;host2:port2”。
linkConfig.password	是	String	连接Redis服务器的密码。
linkConfig.dbIndex	是	String	Redis数据库索引。

### 4.4.1.10 Kafka 连接

#### 介绍

通过Kafka连接器可以与开源的Kafka数据源建立连接，并按照用户指定配置将Kafka中的数据迁移到其它数据源。目前仅支持从Kafka导出数据。

#### 连接样例

```

{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.hadoopType",

```

```

        "value": "MRS"
      },
      {
        "name": "linkConfig.host",
        "value": "192.168.1.147"
      },
      {
        "name": "linkConfig.user",
        "value": "liuhuan1"
      },
      {
        "name": "linkConfig.password",
        "value": "Add password here."
      },
      {
        "name": "linkConfig.authType",
        "value": "KERBEROS"
      }
    ],
    "name": "linkConfig"
  }
},
"extended-configs": {
  "name": "linkConfig.extendedFields",
  "value": "e30="
}
},
"name": "mrs_kafka_link",
"connector-name": "kafka-connector"
}
]
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.hadoopType	是	枚举	Hadoop类型： <ul style="list-style-type: none"> <li>• MRS：表示连接MRS的Kafka。</li> <li>• Apache Kafka：表示连接Apache Kafka的Kafka。</li> </ul>
linkConfig.brokerList	是	String	Apache Kafka 连接需配置此参数。Kafka broker列表，格式如：“host1:port1,host2:port2”。
linkConfig.host	是	String	MRS Manager的浮动IP地址，可以单击输入框后的“选择”来选定已创建的MRS集群，CDM会自动填充下面的鉴权参数。
linkConfig.user	是	String	登录MRS Manager平台的用户名。
linkConfig.password	是	String	登录MRS Manager平台的密码。



参数	是否必选	类型	说明
linkConfig.authType	是	枚举	认证类型，分为以下两种： <ul style="list-style-type: none"> <li>Simple：非安全模式选择 Simple 鉴权。</li> <li>Kerberos：安全模式选择 Kerberos 鉴权。</li> </ul>
linkConfig.properties	否	Map	属性配置，可以添加客户端的配置属性，所添加的每个属性需配置属性名称和值

#### 4.4.1.11 DIS 连接

##### 介绍

通过DIS连接可以与DIS建立连接，并按照用户指定配置将DIS中的数据迁移到其他数据源。

##### 连接样例

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.region",
                "value": "Region"
              },
              {
                "name": "linkConfig.endpoint",
                "value": "https://dis.cn-north-1.myhuaweiclouds.com"
              },
              {
                "name": "linkConfig.ak",
                "value": "RSO6TTEZMJ6TTFBBAACE"
              },
              {
                "name": "linkConfig.sk",
                "value": "Add password here"
              },
              {
                "name": "linkConfig.projectId",
                "value": "11d4d5af17c84660bc90b6631327d7c7"
              }
            ],
            "name": "linkConfig"
          }
        ],
        "name": "dis_link",
        "connector-name": "dis-connector"
      }
    }
  ]
}
```

## 连接参数

参数	是否必选	类型	说明
linkConfig.region	是	String	DIS服务所属地域。
linkConfig.endpoint	是	String	待连接DIS的URL，URL格式为https://Endpoint。
linkConfig.ak	是	String	DIS服务端的AK。
linkConfig.sk	是	String	DIS服务端的SK。
linkConfig.projectId	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 4.4.1.12 Elasticsearch/云搜索服务(CSS)连接

#### 介绍

通过Elasticsearch连接，可以对Elasticsearch服务器或云搜索服务抽取、加载数据。

#### 连接样例

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.host",
                "value": "192.168.0.50:9200;192.168.0.62:9200"
              },
              {
                "name": "linkConfig.safemode",
                "value": "true"
              },
              {
                "name": "linkConfig.user",
                "value": "admin"
              },
              {
                "name": "linkConfig.password",
                "value": "Add password here."
              },
              {
                "name": "linkConfig.linkType",
                "value": "CSS"
              }
            ]
          },
          {
            "name": "linkConfig"
          }
        ],
        "extended-configs": {
          "name": "linkConfig.extendedFields",
          "value": "eyJLodHRwc0FjY2VzcyI6InRydWUifQ=="
        }
      },
      "name": "css-cdm-autotest-nodel",
      "connector-name": "elasticsearch-connector"
    }
  ]
}
```

```
}
]
}
```

## 连接参数

参数	是否必选	类型	说明
linkConfig.host	是	String	配置为Elasticsearch服务器的IP地址或域名，包括端口号，格式为“ip:port”，多个地址之间使用分号(;)分隔，例如：192.168.0.1:9200;192.168.0.2:9200。
linkConfig.safeMode	否	Boolean	当选择安全模式认证时，需要输入用户名、密码和选择是否https访问。
linkConfig.userName	否	String	对于支持用户名密码鉴权的Elasticsearch，需要在创建连接时配置用户名和密码。
linkConfig.password	否	String	登录Elasticsearch的密码。
linkConfig.linkType	是	String	连接类型，用于区分连接的是Elasticsearch或云搜索服务。

### 4.4.1.13 DLI 连接

#### 介绍

通过DLI连接，可以导入数据到数据湖探索（DLI）服务，CDM暂不支持从DLI服务导出数据。

#### 连接样例

```
{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.ak",
                "value": "GRC2WR0IDC6NGROYLWU2"
              },
              {
                "name": "linkConfig.sk",
                "value": "Add password here"
              },
              {
                "name": "linkConfig.region",
                "value": "cn-north-1"
              }
            ],
            {}
          }
        ]
      }
    }
  ]
}
```

```

        "name": "linkConfig.projectId",
        "value": "c48475ce8e174a7a9f775706a3d5ebe2"
    },
    ],
    "name": "linkConfig"
}
]
},
"name": "dli",
"connector-name": "dli-connector"
}
]
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.ak	是	String	登录DLI数据库的AK。
linkConfig.sk	是	String	登录DLI数据库的SK。
linkConfig.region	是	String	DLI服务所在的区域。
linkConfig.projectId	是	String	DLI服务的项目ID。

### 4.4.1.14 DMS Kafka 连接

#### 介绍

通过DMS Kafka连接，可以连接DMS Kafka普通队列或者专享版Kafka。目前仅支持从DMS Kafka导出数据到云搜索服务。

#### 连接样例

```

{
  "links": [
    {
      "link-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "linkConfig.kafkaType",
                "value": "Platinum"
              },
              {
                "name": "linkConfig.brokerList",
                "value": "100.85.121.112:9094,100.85.220.134:9094,100.85.127.232:9094"
              },
              {
                "name": "linkConfig.isPlatinumInstance",
                "value": "false"
              }
            ],
            "name": "linkConfig"
          }
        ],
        "extended-configs": {

```

```

        "name": "linkConfig.extendedFields",
        "value": "e30="
    }
  },
  "name": "dms_kafka",
  "connector-name": "dms-kafka-connector"
}
]
}

```

## 连接参数

参数	是否必选	类型	说明
linkConfig.kafkaType	是	枚举	选择DMS Kafka版本,目前只有专享版。 <ul style="list-style-type: none"> <li>Basic: 指DMS Kafka普通队列。</li> <li>Platinum: 指DMS Kafka专享版。</li> </ul>
linkConfig.brokerList	是	String	DMS Endpoint格式为“host1:port1,host2:port2”。
linkConfig.isPlatinumInstance	是	Boolean	选择是否打开客户端连接Kafka专享版实例时SSL认证的开关。 开启Kafka SASL_SSL, 则数据加密传输, 安全性更高, 但性能会下降。
linkConfig.user	否	String	开启Kafka SASL_SSL时显示该参数, 表示连接DMS Kafka的用户名。
linkConfig.password	否	String	开启Kafka SASL_SSL时显示该参数, 表示连接DMS Kafka的密码。

## 4.4.2 源端作业参数说明

### 4.4.2.1 源端为关系数据库

#### JSON 样例

```

"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.useSql",
          "value": "false"
        }
      ]
    }
  ]
}

```

```

{
  "name": "fromJobConfig.schemaName",
  "value": "rf_database"
},
{
  "name": "fromJobConfig.tableName",
  "value": "rf_from"
},
{
  "name": "fromJobConfig.columnList",
  "value": "AA&BB"
},
{
  "name": "fromJobConfig.incrMigration",
  "value": "false"
}
],
"name": "fromJobConfig"
}
]
}

```

## 参数说明

参数	是否必选	类型	说明
fromJobConfig.useSql	是	Boolean	导出关系型数据库的数据时，可以选择是否使用自定义SQL语句导出。
fromJobConfig.sql	否	String	可以在这里输入自定义的SQL语句，CDM将根据该语句导出数据。
fromJobConfig.schemaName	是	String	<p>数据库模式或表空间，例如：“public”。</p> <p><b>说明</b></p> <p>该参数支持配置通配符（*），实现导出以某一前缀开头或者以某一后缀结尾的所有数据库。例如：</p> <ul style="list-style-type: none"> <li>● <b>SCHEMA*</b>表示导出所有以“SCHEMA”开头的数据库。</li> <li>● <b>*SCHEMA</b>表示导出所有以“SCHEMA”结尾的数据库。</li> <li>● <b>*SCHEMA*</b>表示数据库名称中只要有“SCHEMA”字符串，就全部导出。</li> </ul>

参数	是否必选	类型	说明
fromJobConfig.tableName	是	String	<p>表名，例如：“TBL_EXAMPLE”。</p> <p><b>说明</b> 表名支持配置通配符（*），实现导出以某一前缀开头或者以某一后缀结尾的所有表（要求表中的字段个数和类型都一样）。例如：</p> <ul style="list-style-type: none"> <li>• <b>table*</b>表示导出所有以“table”开头的表。</li> <li>• <b>*table</b>表示导出所有以“table”结尾的表。</li> <li>• <b>*table*</b>表示表名中只要有“table”字符串，就全部导出。</li> </ul>
fromJobConfig.whereClause	否	String	指定抽取的Where子句，不指定则抽取整表，例如：“age > 18 and age <= 60”。
fromJobConfig.columnList	否	String	需要抽取的字段列表，字段名之间使用“&”分割，例如：“id&gid&name”。
fromJobConfig.partitionColumn	否	String	抽取分区字段，依据此字段将作业分割为多个任务并发执行，例如：“id”。
fromJobConfig.usePartition	否	Boolean	从Oracle导出数据时，支持从分区表的各个分区并行抽取数据。启用该功能时，可以通过下面的“fromJobConfig.partitionList”参数指定具体的Oracle表分区，该功能不支持非分区表。
fromJobConfig.partitionList	否	String	输入需要迁移数据的Oracle表分区，多个分区以&分隔，不填则迁移所有分区。

#### 4.4.2.2 源端为对象存储

##### JSON 样例

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.bucketName",
          "value": "cdm-est"
        },
        {
          "name": "fromJobConfig.inputDirectory",
          "value": "/obsfrom/varchar.txt"
        }
      ]
    }
  ]
}
```

```

{
  "name": "fromJobConfig.inputFormat",
  "value": "CSV_FILE"
},
{
  "name": "fromJobConfig.columnList",
  "value": "1&2&3"
},
{
  "name": "fromJobConfig.fieldSeparator",
  "value": ","
},
{
  "name": "fromJobConfig.quoteChar",
  "value": "false"
},
{
  "name": "fromJobConfig.regexSeparator",
  "value": "false"
},
{
  "name": "fromJobConfig.firstRowAsHeader",
  "value": "false"
},
{
  "name": "fromJobConfig.encodeType",
  "value": "UTF-8"
},
{
  "name": "fromJobConfig.fromCompression",
  "value": "NONE"
},
{
  "name": "fromJobConfig.splitType",
  "value": "FILE"
}
],
"name": "fromJobConfig"
}
]
}

```

## 参数说明

参数	是否必选	类型	说明
fromJobConfig.bucketName	是	String	对象存储的桶名。
fromJobConfig.inputDirectory	是	String	抽取文件的路径。支持输入多个文件路径（最多50个），默认以“ ”分隔，也可以自定义文件分隔符，例如：“FROM/example.csv FROM/b.txt”。



参数	是否必选	类型	说明
fromJobConfig.inputFormat	是	枚举	<p>传输数据时所用的文件格式，目前支持以下文件格式：</p> <ul style="list-style-type: none"> <li>• CSV_FILE: CSV格式，用于迁移文件到数据表的场景。</li> <li>• JSON_FILE: JSON格式，一般都是用于迁移文件到数据表的场景。</li> <li>• BINARY_FILE: 二进制格式，不解析文件内容直接传输，不要求文件格式必须为二进制。适用于文件到文件的原样复制。</li> </ul> <p>当选择“BINARY_FILE”时，目的端也必须为文件系统。</p>
fromJobConfig.lineSeparator	否	String	<p>文件中的换行符，默认自动识别“\n”、“\r”或“\r\n”。手动配置特殊字符，如空格回车需使用URL编码后的值。或通过编辑作业json方式配置，无需URL编码。</p>
fromJobConfig.columnList	否	String	<p>需要抽取的列号，列号之间使用“&amp;”分割，并由小到大排序，例如：“1&amp;3&amp;5”。</p>
fromJobConfig.regexSeparator	否	Boolean	<p>是否使用正则表达式分割字段，当文件格式为“CSV_FILE”时此参数有效。</p>
fromJobConfig.regex	否	String	<p>正则表达式，当选择使用正则表达式分割字段时，此参数有效。</p>
fromJobConfig.fieldSeparator	否	String	<p>字段分隔符，当文件格式为“CSV_FILE”时此参数有效，默认值为：“，”。</p>
fromJobConfig.quoteChar	否	Boolean	<p>是否使用包围符，选择“true”时，包围符内的字段分隔符会被视为字符串值的一部分，目前CDM默认的包围符为：“”。</p>
fromJobConfig.firstRowAsHeader	否	Boolean	<p>是否默认首行为标题行，当文件格式为“CSV_FILE”时此参数有效。在迁移CSV文件到表时，CDM默认是全部写入，当该参数选择“true”时，CDM会将CSV文件的第一行数据作为标题行，不写入目的端的表。</p>

参数	是否必选	类型	说明
fromJobConfig.fromCompression	否	枚举	<p>压缩格式，当文件格式为“CSV_FILE”或“JSON”时此参数有效。选择对应压缩格式的源文件：</p> <ul style="list-style-type: none"> <li>• NONE：表示传输所有格式的文件。</li> <li>• GZIP：表示只传输GZIP格式的文件。</li> <li>• ZIP：表示只传输ZIP格式的文件。</li> </ul>
fromJobConfig.jsonReferenceNode	否	String	<p>记录节点，当文件格式为“JSON_FILE”时此参数有效。对该JSON节点下的数据进行解析，如果该节点对应的数据为JSON数组，那么系统会以同一模式从该数组中提取数据。多层嵌套的JSON节点以字符“.”分割，例如：“data.list”。</p>
fromJobConfig.encodingType	否	String	<p>编码类型，例如：“UTF_8”或“GBK”。</p>
fromJobConfig.useMarkerFile	否	Boolean	<p>选择是否开启作业标识文件的功能。当源端路径下存在启动作业的标识文件时才启动作业，否则会挂起等待一段时间，等待时长在下方 “fromJobConfig.waitTime”（等待时间）参数中配置。</p>
fromJobConfig.markerFile	否	String	<p>启动作业的标识文件名。指定文件后，只有在源端路径下存在该文件的情况下才会运行任务，不指定时默认不启用该功能，例如：“ok.txt”。</p>
fromJobConfig.waitTime	否	String	<p>选择开启作业标识文件的功能时，如果源路径下不存在启动作业的标识文件，作业挂机等待的时长，当超时后任务会失败。 等待时间设置为0时，当源端路径下不存在标识文件，任务会立即失败。 单位：秒。</p>

参数	是否必选	类型	说明
fromJobConfig.filterType	否	枚举	选择过滤器类型： <ul style="list-style-type: none"> <li>WILDCARD: 输入通配符作为过滤文件，满足过滤条件的路径或文件会被传输。</li> <li>TIME: 时间过滤器，当文件的修改时间晚于输入的时间时，该文件才会被传输。</li> </ul>
fromJobConfig.pathFilter	否	String	路径过滤器，过滤类型为通配符时配置，用于过滤文件目录，例如：“*input”。
fromJobConfig.fileFilter	否	String	文件过滤器，过滤类型为通配符时配置，用于过滤目录下的文件，支持配置多个文件，中间使用“,”分隔，例如：“*.csv,*.txt”。
fromJobConfig.startTime	否	String	“过滤类型”选择“时间过滤器”时，可以指定一个时间值，当文件的修改时间大于等于该时间才会被传输，输入的时间格式需为“yyyy-MM-dd HH:mm:ss”。 该参数支持配置为时间宏变量，例如\$ <b>{timestamp(dateformat(yyyy-MM-dd HH:mm:ss,-90,DAY))}</b> 表示：只迁移最近90天内的文件。
fromJobConfig.endTime	否	String	“过滤类型”选择“时间过滤器”时，可以指定一个时间值，当文件的修改时间小于该时间才会被传输，输入的时间格式需为“yyyy-MM-dd HH:mm:ss”。 该参数支持配置为时间宏变量，例如\$ <b>{timestamp(dateformat(yyyy-MM-dd HH:mm:ss))}</b> 表示：只迁移修改时间为当前时间以前的文件。
fromJobConfig.fileSeparator	否	String	“fromJobConfig.inputDirectory”（抽取文件的路径）参数中如果输入的是多个文件路径，CDM使用这里配置的文件分隔符来区分各个文件，默认为“ ”。

参数	是否必选	类型	说明
fromJobConfig.md5FileSuffix	否	String	校验CDM抽取的文件，是否与源文件一致。

### 4.4.2.3 源端为 HDFS

#### JSON 样例

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.inputDirectory",
          "value": "/hdfsfrom/from_hdfs_est.csv"
        },
        {
          "name": "fromJobConfig.inputFormat",
          "value": "CSV_FILE"
        },
        {
          "name": "fromJobConfig.columnList",
          "value": "1"
        },
        {
          "name": "fromJobConfig.fieldSeparator",
          "value": ","
        },
        {
          "name": "fromJobConfig.quoteChar",
          "value": "false"
        },
        {
          "name": "fromJobConfig.regexSeparator",
          "value": "false"
        },
        {
          "name": "fromJobConfig.firstRowAsHeader",
          "value": "false"
        },
        {
          "name": "fromJobConfig.encodeType",
          "value": "UTF-8"
        },
        {
          "name": "fromJobConfig.fromCompression",
          "value": "NONE"
        },
        {
          "name": "fromJobConfig.compressedFileSuffix",
          "value": ""
        },
        {
          "name": "fromJobConfig.splitType",
          "value": "FILE"
        },
        {
          "name": "fromJobConfig.useMarkerFile",
          "value": "false"
        },
        {
          "name": "fromJobConfig.fileSeparator",
          "value": "|"
        }
      ]
    }
  ]
}
```

```

    {
      "name": "fromJobConfig.filterType",
      "value": "NONE"
    }
  ],
  "name": "fromJobConfig"
}
]
}

```

## 参数说明

- HDFS作业参数说明

参数	是否必选	类型	说明
fromJobConfig.inputDirectory	是	String	待抽取数据的路径，例如：“/data_dir”。
fromJobConfig.inputFormat	是	枚举	传输数据时所用的文件格式，目前支持以下文件格式： <ul style="list-style-type: none"> <li>• CSV_FILE: CSV格式。</li> <li>• PARQUET_FILE: PARQUET格式。</li> <li>• BINARY_FILE: 二进制格式。</li> </ul> 当选择“BINARY_FILE”时，目的端也必须为文件系统。
fromJobConfig.columnList	否	String	需要抽取的列号，列号之间使用“&”分割，并由小到大排序，例如：“1&3&5”。
fromJobConfig.lineSeparator	否	String	文件中的换行符，默认自动识别“\n”、“\r”或“\r\n”。手动配置特殊字符，如空格回车需使用URL编码后的值。或通过编辑作业json方式配置，无需URL编码。
fromJobConfig.fieldSeparator	否	String	字段分隔符，当文件格式为“CSV_FILE”时此参数有效，默认值为：“，”。
fromJobConfig.quoteChar	否	Boolean	是否使用包围符，选择“true”时，包围符内的字段分隔符会被视为字符串值的一部分，目前CDM默认的包围符为：“”。
fromJobConfig.regexSeparator	否	Boolean	是否使用正则表达式分割字段，当文件格式为“CSV_FILE”时此参数有效。
fromJobConfig.encodeType	否	String	编码类型，例如：“UTF_8”或“GBK”。

参数	是否必选	类型	说明
fromJobConfig.firstRowAsHeader	否	Boolean	是否默认首行为标题行，当文件格式为“CSV_FILE”时此参数有效。在迁移CSV文件到表时，CDM默认是全部写入，当该参数选择“true”时，CDM会将CSV文件的第一行数据作为标题行，不写入目的端的表。
fromJobConfig.fromCompression	否	枚举	压缩格式，表示选择只传输对应压缩格式的源文件。“NONE”表示传输所有格式的文件。
fromJobConfig.compressedFileSuffix	否	String	需要解压缩的文件后缀名。当一批文件中以该值为后缀时，才会执行解压缩操作，否则保持原样传输。当输入*或为空时，所有文件都会被解压。
fromJobConfig.splitType	否	枚举	指定任务分片方式，选择按文件或文件大小进行分割。HDFS上的文件，如果在HDFS上已经分片，则HDFS每个分片视为一个文件。 <ul style="list-style-type: none"> <li>• FILE：按文件数量进行分片。例如有10个文件，并在任务参数中指定“throttlingConfig.numExtractors”（抽取并发数）为“5”，则每个分片2个文件。</li> <li>• SIZE：按文件大小分割。注意这里不会将文件做切分来实现均衡。例如：有10个文件，9个10M，1个200M，在并发任务数中指定“throttlingConfig.numExtractors”（抽取并发数）为“2”，则会分两个分片，一个处理9个10M的文件，一个处理1个200M的文件。</li> </ul>

参数	是否必选	类型	说明
fromJobConfig.useMarkerFile	否	Boolean	选择是否开启作业标识文件的功能。当源端路径下存在启动作业的标识文件时才启动作业，否则会挂起等待一段时间，等待时长在下方“fromJobConfig.waitTime”（等待时间）参数中配置。
fromJobConfig.markerFile	否	String	启动作业的标识文件名。指定文件后，只有在源端路径下存在该文件的情况下才会运行任务，不指定时默认不启用该功能，例如：“ok.txt”。
fromJobConfig.fileSeparator	否	String	“fromJobConfig.inputDirectory”（抽取文件的路径）参数中如果输入的是多个文件路径，CDM使用这里配置的文件分隔符来区分各个文件，默认为“ ”。
fromJobConfig.filterType	否	枚举	选择过滤器类型： <ul style="list-style-type: none"> <li>• WILDCARD：输入通配符作为过滤文件，满足过滤条件的路径或文件会被传输。</li> <li>• TIME：时间过滤器，当文件的修改时间晚于输入的时间时，该文件才会被传输。</li> </ul>
fromJobConfig.pathFilter	否	String	路径过滤器，过滤类型为通配符时配置，用于过滤文件目录，例如：“*input”。
fromJobConfig.fileFilter	否	String	文件过滤器，过滤类型为通配符时配置，用于过滤目录下的文件，支持配置多个文件，中间使用“,”分隔，例如：“*.csv,*.txt”。

参数	是否必选	类型	说明
fromJobConfig. startTime	否	String	<p>“过滤类型”选择“时间过滤器”时，可以指定一个时间值，当文件的修改时间大于等于该时间才会被传输，输入的时间格式需为“yyyy-MM-dd HH:mm:ss”。</p> <p>该参数支持配置为时间宏变量，例如\$ <b>{timestamp(dateformat(yy yy-MM-dd HH:mm:ss,-90,DAY))}</b>表示：只迁移最近90天内的文件。</p>
fromJobConfig. endTime	否	String	<p>“过滤类型”选择“时间过滤器”时，可以指定一个时间值，当文件的修改时间小于该时间才会被传输，输入的时间格式需为“yyyy-MM-dd HH:mm:ss”。</p> <p>该参数支持配置为时间宏变量，例如\$ <b>{timestamp(dateformat(yy yy-MM-dd HH:mm:ss))}</b>表示：只迁移修改时间为当前时间以前的文件。</p>
fromJobConfig. createSnapshot	否	Boolean	<p>如果配置为“true”，CDM读取HDFS系统上的文件时，会先对待迁移的源目录创建快照（不允许对单个文件创建快照），然后CDM迁移快照中的数据。</p> <p>需要HDFS系统的管理员权限才可以创建快照，CDM作业完成后，快照会被删除。</p>
fromJobConfig. formats	否	数据结构	<p>时间格式，当“fromJobConfig.inputFormat”（文件格式）为“CSV_FILE”（CSV格式），并且文件中有时间类型字段时，才需要输入，具体说明请参见<a href="#">fromJobConfig.formats</a>参数说明。</p>



参数	是否必选	类型	说明
fromJobConfig.deryption	否	枚举	<p>“fromJobConfig.inputFormat”（文件格式）选择为“BINARY_FILE”（二进制格式）时才有该参数，选择是否对已加密的文件解密后再导出，以及解密方式：</p> <ul style="list-style-type: none"> <li>• NONE：不解密，直接导出文件。</li> <li>• AES-256-GCM：使用AES-256-GCM（NoPadding）算法解密后再导出文件。</li> </ul>
fromJobConfig.dek	否	String	<p>数据解密密钥，密钥由长度64的十六进制数组成，且必须与加密时配置的“toJobConfig.dek”（导入时配置的数据加密密钥）一致。如果不一致系统不会报异常，只是解密出来的数据会错误。</p>
fromJobConfig.iv	否	String	<p>解密需要的初始化向量，初始化向量由长度32的十六进制数组成，且必须与加密时配置的“toJobConfig.iv”（导入时配置的初始化向量）一致。如果不一致系统不会报异常，只是解密出来的数据会错误。</p>

- fromJobConfig.formats参数说明

参数	是否必选	类型	说明
name	是	String	列号，例如：“1”。
value	是	String	时间格式，例如：“yyyy-MM-dd”。

#### 4.4.2.4 源端为 Hive

##### JSON 样例

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.hive",
          "value": "hive"
        }
      ],
      {
```

```

    "name": "fromJobConfig.database",
    "value": "rf_database"
  },
  {
    "name": "fromJobConfig.table",
    "value": "rf_from"
  },
  {
    "name": "fromJobConfig.columnList",
    "value": "tiny&small&int&integer&bigint&float&double&timestamp&char&varchar&text"
  }
],
"name": "fromJobConfig"
}
]
}

```

## 参数说明

参数	是否必选	类型	说明
fromJobConfig.hive	否	String	待抽取数据的数据源，作业源端为 Hive 时，这里为“hive”。
fromJobConfig.database	否	String	待抽取数据的数据库，例如“default”。
fromJobConfig.table	是	String	待抽取数据的表名，例如“cdm”。
fromJobConfig.columnList	否	String	需要抽取的列号，列号之间使用“&”分割，并由小到大排序，例如：“1&3&5”。

### 4.4.2.5 源端为 HBase/CloudTable

#### JSON 样例

```

"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.table",
          "value": "rf_from"
        },
        {
          "name": "fromJobConfig.columnFamilies",
          "value": "rowkey&f"
        },
        {
          "name": "fromJobConfig.columns",
          "value": "rowkey.rowkey&f_small"
        },
        {
          "name": "fromJobConfig.formats",
          "value": {
            "f_date": "yyyy-MM-dd",
            "f_timestamp": "yyyy-MM-dd HH:mm:ss"
          }
        }
      ]
    }
  ],
}

```

```

    "name": "fromJobConfig"
  }
]
}

```

## 参数说明

- HBase/CloudTable作业参数说明

参数	是否必选	类型	说明
fromJobConfig.table	是	String	需要抽取数据的表名，例如“cdm”。
fromJobConfig.columnFamilies	否	String	抽取数据所属的列族。
fromJobConfig.columns	否	String	需要抽取的列，列号之间使用“&”分割，列族与列之间用“:”分隔，例如：“cf1:c1&cf2:c2”。
fromJobConfig.isSplit	否	Boolean	选择是否拆分Rowkey，例如“true”。
fromJobConfig.delimiter	否	String	用于切分Rowkey的分隔符，若不设置则不切分，例如“ ”。
fromJobConfig.startTime	否	String	时间区间左边界（包含该值），格式为“yyyy-MM-dd hh:mm:ss”。 表示只抽取该时间及以后的数据。
fromJobConfig.endTime	否	String	时间区间右边界（不包含该值），格式为“yyyy-MM-dd hh:mm:ss”。 表示只抽取该时间以前的数据。
fromJobConfig.formats	否	数据结构	时间格式，请参见 <a href="#">fromJobConfig.formats参数说明</a> 。

- fromJobConfig.formats参数说明

参数	是否必选	类型	说明
name	是	String	列号，例如：“1”。
value	是	String	时间格式，例如：“yyyy-MM-dd”。

## 4.4.2.6 源端为 FTP/SFTP

### JSON 样例

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.inputDirectory",
          "value": "/sftpfrom/from_sftp.csv"
        },
        {
          "name": "fromJobConfig.inputFormat",
          "value": "CSV_FILE"
        },
        {
          "name": "fromJobConfig.columnList",
          "value": "1&2&3&4&5&6&7&8&9&10&11&12"
        },
        {
          "name": "fromJobConfig.fieldSeparator",
          "value": ","
        },
        {
          "name": "fromJobConfig.regexSeparator",
          "value": "false"
        },
        {
          "name": "fromJobConfig.firstRowAsHeader",
          "value": "false"
        },
        {
          "name": "fromJobConfig.encodeType",
          "value": "UTF-8"
        },
        {
          "name": "fromJobConfig.fromCompression",
          "value": "NONE"
        },
        {
          "name": "fromJobConfig.splitType",
          "value": "FILE"
        }
      ],
      "name": "fromJobConfig"
    }
  ]
}
```

### 参数说明

FTP、SFTP的源端作业参数相同，如表4-204所示。

表 4-204 文件类源端作业参数

参数	是否必选	类型	说明
fromJobConfig.inputDirectory	是	String	抽取文件的路径。支持输入多个文件路径（最多50个），默认以“ ”分隔，也可以自定义文件分隔符，例如：“FROM/example.csv FROM/b.txt”。

参数	是否必选	类型	说明
fromJobConfig.inputFormat	是	枚举	<p>传输数据时所用的文件格式，目前支持以下文件格式：</p> <ul style="list-style-type: none"> <li>• CSV_FILE: CSV格式，用于迁移文件到数据表的场景。</li> <li>• JSON_FILE: JSON格式，一般都是用于迁移文件到数据表的场景。</li> <li>• BINARY_FILE: 二进制格式，不解析文件内容直接传输，不要求文件格式必须为二进制。适用于文件到文件的原样复制。</li> </ul> <p>当选择“BINARY_FILE”时，目的端也必须为文件系统。</p>
fromJobConfig.lineSeparator	否	String	<p>文件中的换行符，默认自动识别“\n”、“\r”或“\r\n”。手动配置特殊字符，如空格回车需使用URL编码后的值。或通过编辑作业json方式配置，无需URL编码。</p>
fromJobConfig.columnList	否	String	<p>需要抽取的列号，列号之间使用“&amp;”分割，并由小到大排序，例如：“1&amp;3&amp;5”。</p>
fromJobConfig.fieldSeparator	否	String	<p>字段分隔符，当文件格式为“CSV_FILE”时此参数有效，默认值为：“，”。</p>
fromJobConfig.quoteChar	否	Boolean	<p>是否使用包围符，选择“true”时，包围符内的字段分隔符会被视为字符串值的一部分，目前CDM默认的包围符为：“”。</p>
fromJobConfig.regexSeparator	否	Boolean	<p>是否使用正则表达式分割字段，当文件格式为“CSV_FILE”时此参数有效。</p>
fromJobConfig.regex	否	String	<p>正则表达式，当选择使用正则表达式分割字段时，此参数有效。</p>
fromJobConfig.firstRowAsHeader	否	Boolean	<p>是否默认首行为标题行，当文件格式为“CSV_FILE”时此参数有效。在迁移CSV文件到表时，CDM默认是全部写入，当该参数选择“true”时，CDM会将CSV文件的第一行数据作为标题行，不写入目的端的表。</p>

参数	是否必选	类型	说明
fromJobConfig.fromCompression	否	枚举	<p>压缩格式，当文件格式为“CSV_FILE”或“JSON”时此参数有效。选择对应压缩格式的源文件：</p> <ul style="list-style-type: none"> <li>• NONE：表示传输所有格式的文件。</li> <li>• GZIP：表示只传输GZIP格式的文件。</li> <li>• ZIP：表示只传输ZIP格式的文件。</li> </ul>
fromJobConfig.splitType	否	枚举	<p>指定任务分片方式，选择按文件或文件大小进行分割。</p> <ul style="list-style-type: none"> <li>• FILE：按文件数量进行分片。例如有10个文件，并在任务参数中指定“throttlingConfig.numExtractors”（抽取并发数）为“5”，则每个分片2个文件。</li> <li>• SIZE：按文件大小分割。注意这里不会将文件做切分来实现均衡。例如：有10个文件，9个10M，1个200M，在并发任务数中指定“throttlingConfig.numExtractors”（抽取并发数）为“2”，则会分两个分片，一个处理9个10M的文件，一个处理1个200M的文件。</li> </ul>
fromJobConfig.jsonReferenceNode	否	String	<p>记录节点，当文件格式为“JSON_FILE”时此参数有效。对该JSON节点下的数据进行解析，如果该节点对应的数据为JSON数组，那么系统会以同一模式从该数组中提取数据。多层嵌套的JSON节点以字符“.”分割，例如：“data.list”。</p>
fromJobConfig.encodedType	否	String	<p>编码类型，例如：“UTF_8”或“GBK”。</p>
fromJobConfig.useMarkerFile	否	Boolean	<p>选择是否开启作业标识文件的功能。当源端路径下存在启动作业的标识文件时才启动作业，否则会挂起等待一段时间，等待时长在下方“fromJobConfig.waitTime”（等待时间）参数中配置。</p>

参数	是否必选	类型	说明
fromJobConfig.markerFile	否	String	启动作业的标识文件名。指定文件后，只有在源端路径下存在该文件的情况下才会运行任务，不指定时默认不启用该功能，例如：“ok.txt”。
fromJobConfig.waitTime	否	String	选择开启作业标识文件的功能时，如果源路径下不存在启动作业的标识文件，作业挂机等待的时长，当超时后任务会失败。 等待时间设置为0时，当源端路径下不存在标识文件，任务会立即失败。 单位：秒。
fromJobConfig.filterType	否	枚举	选择过滤器类型： <ul style="list-style-type: none"> <li>● WILDCARD：输入通配符作为过滤文件，满足过滤条件的路径或文件会被传输。</li> <li>● TIME：时间过滤器，当文件的修改时间晚于输入的时间时，该文件才会被传输。</li> </ul>
fromJobConfig.pathFilter	否	String	路径过滤器，过滤类型为通配符时配置，用于过滤文件目录，例如：“*input”。
fromJobConfig.fileFilter	否	String	文件过滤器，过滤类型为通配符时配置，用于过滤目录下的文件，支持配置多个文件，中间使用“,”分隔，例如：“*.csv,*.txt”。
fromJobConfig.startTime	否	String	“过滤类型”选择“时间过滤器”时，可以指定一个时间值，当文件的修改时间大于等于该时间才会被传输，输入的时间格式需为“yyyy-MM-dd HH:mm:ss”。 该参数支持配置为时间宏变量，例如\$ <b>{timestamp(dateformat(yyy y-MM-dd HH:mm:ss,-90,DAY))}</b> 表示： 只迁移最近90天内的文件。

参数	是否必选	类型	说明
fromJobConfig.endTime	否	String	“过滤类型”选择“时间过滤器”时，可以指定一个时间值，当文件的修改时间小于该时间才会被传输，输入的时间格式需为“yyyy-MM-dd HH:mm:ss”。该参数支持配置为时间宏变量，例如\${timestamp(dateformat(yyy-MM-dd HH:mm:ss))}表示：只迁移修改时间为当前时间以前的文件。
fromJobConfig.fileSeparator	否	String	“fromJobConfig.inputDirectory”（抽取文件的路径）参数中如果输入的是多个文件路径，CDM使用这里配置的文件分隔符来区分各个文件，默认为“ ”。
fromJobConfig.md5FileSuffix	否	String	校验CDM抽取的文件，是否与源文件一致。

#### 4.4.2.7 源端为 HTTP/HTTPS

##### JSON 样例

```

"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.inputDirectory",
          "value": "http://10.114.196.186:8080/httpfrom/symbol.txt"
        },
        {
          "name": "fromJobConfig.inputFormat",
          "value": "BINARY_FILE"
        },
        {
          "name": "fromJobConfig.fromCompression",
          "value": "TARGZ"
        },
        {
          "name": "fromJobConfig.compressedFileSuffix",
          "value": ""
        },
        {
          "name": "fromJobConfig.fileSeparator",
          "value": "|"
        }
      ],
      "name": "fromJobConfig"
    }
  ]
}

```



## 参数说明

参数	是否必选	类型	说明
fromJobConfig.inputDirectory	是	String	待抽取文件的URL。 用于读取一个公网HTTP/HTTPS URL的文件，包括第三方对象存储的公共读取场景和网盘场景。
fromJobConfig.inputFormat	是	枚举	传输数据时所用的文件格式，目前只支持二进制格式。
fromJobConfig.fromCompression	否	枚举	选择对应压缩格式的源文件进行迁移： <ul style="list-style-type: none"> <li>• NONE：表示传输所有格式的文件。</li> <li>• GZIP：表示只传输GZIP格式的文件。</li> <li>• ZIP：表示只传输ZIP格式的文件。</li> <li>• TAR.GZ：表示只传输TAR.GZ格式的文件。</li> </ul>
fromJobConfig.compressedFileSuffix	否	String	需要解压缩的文件后缀名。当一批文件中以该值为后缀时，才会执行解压缩操作，否则保持原样传输。当输入*或为空时，所有文件都会被解压。
fromJobConfig.fileSeparator	否	String	传输多个文件时，CDM使用这里配置的文件分隔符来区分各个文件，默认为 。
fromJobConfig.useQuery	否	Boolean	<ul style="list-style-type: none"> <li>• 该参数设置为“true”时，上传到OBS的对象使用的对象名，为去掉query参数后的字符。</li> <li>• 该参数设置为“false”时，上传到OBS的对象使用的对象名，包含query参数。</li> </ul>
fromJobConfig.md5FileSuffix	否	String	校验CDM抽取的文件，是否与源文件一致。

### 4.4.2.8 源端为 MongoDB/DDS

#### JSON 样例

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
```

```

{
  "name": "fromJobConfig.database",
  "value": "cdm"
},
{
  "name": "fromJobConfig.collectionName",
  "value": "rf_from"
},
{
  "name": "fromJobConfig.columnList",
  "value": "TINYTEST&SMALLTEST&INTTEST&INTEGERTEST&BIGINTTEST&FLOATTEST"
},
{
  "name": "fromJobConfig.isBatchMigration",
  "value": "false"
},
{
  "name": "fromJobConfig.filters",
  "value": "{ 'last_name': 'Smith' }"
}
],
"name": "fromJobConfig"
}
]
}

```

## 参数说明

参数	是否必选	类型	说明
fromJobConfig.database	是	String	MongoDB/DDS的数据库名称。
fromJobConfig.collectionName	是	String	MongoDB/DDS的集合名称。
fromJobConfig.columnList	否	String	需要抽取的字段列表，字段名之间使用“&”分割，例如：“id&gid&name”。
fromJobConfig.isBatchMigration	否	Boolean	是否为整库迁移。
fromJobConfig.filters	否	String	创建用于匹配文档的筛选条件，CDM只迁移符合条件的数据。例如： <ol style="list-style-type: none"> <li>按表达式对象筛选：例如 { 'last_name': 'Smith' }，表示查找所有“last_name”属性值为“Smith”的文档。</li> <li>按参数选项筛选：例如 { x: "john" }, { z: 1 }，表示查找 x=john的所有z字段。</li> <li>按条件筛选：例如 { "field": { \$gt: 5 } }，表示查找field字段中大于5的值。</li> </ol>

### 4.4.2.9 源端为 Redis

#### JSON 样例

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.isBatchMigration",
          "value": "false"
        },
        {
          "name": "fromJobConfig.keyPrefix",
          "value": "rf_string_from"
        },
        {
          "name": "fromJobConfig.keySeparator",
          "value": ":"
        },
        {
          "name": "fromJobConfig.valueStoreType",
          "value": "STRING"
        },
        {
          "name": "fromJobConfig.valueSeparator",
          "value": ","
        },
        {
          "name": "fromJobConfig.columnList",
          "value": "1&2&3&4&5&6&7&8&9&10&11&12"
        }
      ],
      "name": "fromJobConfig"
    }
  ]
}
```

#### 参数说明

- Redis源端作业参数说明

参数	是否必选	类型	说明
fromJobConfig.isBatchMigration	否	Boolean	是否为整库迁移。
fromJobConfig.keyPrefix	是	String	key前缀，对应关系表的表名。 Redis和关系表的映射：用关系表的“表名+分隔符”来对应Redis的Key；关系表的一行数据对应Redis的Value。
fromJobConfig.keySeparator	是	String	key分隔符，一般用来分割关系表和主键。

参数	是否必选	类型	说明
fromJobConfig.valueStoreType	是	String	关系表行数据在Redis中的存储方式分为“string”和“hash”两种存储方式。 <ul style="list-style-type: none"> <li>• STRING: 表示用字符串通过分隔符来表示一行数据的各列，可以有效节省存储空间。</li> <li>• HASH: 表示一行数据通过“列名: 列值”的方式存储在hash表中。</li> </ul>
fromJobConfig.valueSeparator	否	String	值分隔符号，当“valueStoreType”为“STRING”时此参数有效，默认值为：“\tab”。
fromJobConfig.columnList	否	String	需要抽取的字段列表，字段名之间使用“&”分割，例如：“id&gid&name”。
fromJobConfig.formats	否	数据结构	时间格式，请参见 <a href="#">fromJobConfig.formats参数说明</a> 。

- fromJobConfig.formats参数说明

参数	是否必选	类型	说明
name	是	String	列号，例如：“1”。
value	是	String	时间格式，例如：“yyyy-MM-dd”。

#### 4.4.2.10 源端为 DIS

##### JSON 样例

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.streamName",
          "value": "cdm"
        },
        {
          "name": "fromJobConfig.disConsumerStrategy",
          "value": "FROM_LAST_STOP"
        },
        {
          "name": "fromJobConfig.isPermanency",
          "value": "true"
        }
      ]
    }
  ]
}
```

```

    {
      "name": "fromJobConfig.maxPollRecords",
      "value": "100"
    },
    {
      "name": "fromJobConfig.shardId",
      "value": "0"
    },
    {
      "name": "fromJobConfig.dataFormat",
      "value": "BINARY"
    },
    {
      "name": "fromJobConfig.separator",
      "value": ","
    }
  ],
  "name": "fromJobConfig"
}
]
}

```

## 参数说明

参数	是否必选	类型	说明
fromJobConfig.streamName	是	String	DIS的通道名。
fromJobConfig.disConsumerStrategy	是	枚举	设置从DIS拉取数据时的初始偏移量： <ul style="list-style-type: none"> <li>• LATEST：最大偏移量，即最新的数据。</li> <li>• FROM_LAST_STOP：从上次停止处继续拉取。</li> <li>• EARLIEST：最小偏移量，即最早的数据。</li> </ul>
fromJobConfig.isPermanency	是	Boolean	是否永久运行。
fromJobConfig.maxPollRecords	否	String	每次向DIS请求数据限制最大请求记录数。
fromJobConfig.shardId	是	String	DIS分区ID，该参数支持输入多个分区ID，使用“,”分隔。
fromJobConfig.dataFormat	是	枚举	解析数据时使用的格式： <ul style="list-style-type: none"> <li>• BINARY：适用于文件迁移场景，不解析数据内容原样传输。</li> <li>• CSV：以CSV格式解析源数据。</li> </ul>
fromJobConfig.separator	否	String	字段分隔符。

参数	是否必选	类型	说明
fromJobConfig.appName	否	String	用户数据消费程序的唯一标识符。

#### 4.4.2.11 源端为 Kafka

##### JSON 样例

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.topicsList",
          "value": "est1,est2"
        },
        {
          "name": "fromJobConfig.kafkaConsumerStrategy",
          "value": "EARLIEST"
        },
        {
          "name": "fromJobConfig.isPermanency",
          "value": "true"
        }
      ],
      "name": "fromJobConfig"
    }
  ]
}
```

##### 参数说明

参数	是否必选	类型	说明
fromJobConfig.topicsList	是	String	Kafka topic列表，可以为多个 topic，以“,”作为分隔符。
fromJobConfig.kafkaConsumerStrategy	是	枚举	从Kafka拉取数据时的初始偏移量设置： <ul style="list-style-type: none"> <li>• LATEST：最大偏移量，即最新的数据。</li> <li>• EARLIEST：最小偏移量，即最老的数据。</li> </ul>
fromJobConfig.isPermanency	是	Boolean	是否永久运行。
fromJobConfig.groupId	否	String	用户指定消费组ID。 如果是从DMS Kafka导出数据，专享版请任意输入，标准版请输入有效的消费组ID。

参数	是否必选	类型	说明
fromJobConfig.dataFormat	是	枚举	解析数据时使用的格式： <ul style="list-style-type: none"> <li>• BINARY：适用于文件迁移场景，不解析数据内容原样传输。</li> <li>• CSV：以CSV格式解析源数据。</li> </ul>
fromJobConfig.maxPollRecords	否	String	每次向Kafka请求数据限制最大请求记录数。
fromJobConfig.maxPollInterval	否	String	每次poll之间的最大时间间隔。
fromJobConfig.separator	否	String	字段分隔符。

#### 4.4.2.12 源端为 Elasticsearch/云搜索服务

##### JSON 样例

```

"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.index",
          "value": "cdm"
        },
        {
          "name": "fromJobConfig.type",
          "value": "es"
        },
        {
          "name": "fromJobConfig.columnList",
          "value": "a1:numeric&s1:string"
        },
        {
          "name": "fromJobConfig.splitNestedField",
          "value": "true"
        },
        {
          "name": "fromJobConfig.queryString",
          "value": "last_name:Smith"
        }
      ],
      "name": "fromJobConfig"
    }
  ]
}

```

## 参数说明

参数	是否必选	类型	说明
fromJobConfig.index	是	String	抽取数据的索引，类似关系数据库中的数据库名称。
fromJobConfig.type	是	String	抽取数据的类型，类似关系数据库中的表名。
fromJobConfig.columnList	否	String	需要抽取的字段列表，字段名之间使用“&”分隔，例如：“id&gid&name”。
fromJobConfig.splitNestedField	否	Boolean	选择是否将nested字段的json内容拆分，例如：将“a:{ b:{ c:1, d:{ e:2, f:3 } } }”拆成三个字段“a.b.c”、“a.b.d.e”、“a.b.d.f”。
fromJobConfig.queryString	否	String	使用Elasticsearch的查询字符串（query string）对源数据进行过滤，CDM只迁移满足过滤条件的数据。

### 4.4.3 目的端作业参数说明

#### 4.4.3.1 目的端为关系数据库

##### JSON 样例

```
"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.schemaName",
          "value": "cdm"
        },
        {
          "name": "toJobConfig.tablePreparation",
          "value": "DROP_AND_CREATE"
        },
        {
          "name": "toJobConfig.tableName",
          "value": "rf_to"
        },
        {
          "name": "toJobConfig.columnList",
          "value": "id&gid&name"
        },
        {
          "name": "toJobConfig.isCompress",
          "value": "false"
        },
        {
          "name": "toJobConfig.orientation",
          "value": "ROW"
        }
      ]
    }
  ]
}
```



```

    },
    {
      "name": "toJobConfig.useStageTable",
      "value": "false"
    },
    {
      "name": "toJobConfig.shouldClearTable",
      "value": "false"
    },
    {
      "name": "toJobConfig.extendCharLength",
      "value": "false"
    }
  ],
  "name": "toJobConfig"
}
]
}

```

## 参数说明

参数	是否必选	类型	说明
toJobConfig.schemaName	是	String	数据库模式或表空间。
toJobConfig.tablePreparation	是	枚举	只有当源端和目的端都为关系数据库时，才有该参数。表示写入表数据时，用户选择的操作： <ul style="list-style-type: none"> <li>• DO_NOTHING：不自动建表。</li> <li>• CREATE_WHEN_NOT_EXIST：当目的端的数据库没有“tableName”参数中指定的表时，CDM会自动创建该表。</li> <li>• DROP_AND_CREATE：先删除“tableName”参数中指定的表，然后再重新创建该表。</li> </ul>
toJobConfig.tableName	是	String	写入数据的表名。
toJobConfig.columnList	否	String	需要加载的字段列表，字段名之间使用“&”分割，例如：“id&gid&name”。

参数	是否必选	类型	说明
toJobConfig.beforeImportType	否	枚举	<p>导入数据前，选择是否清除目的表的数据：</p> <ul style="list-style-type: none"> <li>• none：写入数据前不清除目标表中数据，数据追加写入。</li> <li>• shouldClearTable：写入数据前会清除目标表中数据。</li> <li>• whereClause：选择根据where条件删除时，需要配置“toJobConfig.whereClause”参数，CDM根据条件选择性删除目标表的数据。</li> </ul>
toJobConfig.whereClause	否	String	where条件，导入前根据where条件删除目的表的数据。
toJobConfig.orientation	否	枚举	<p>存储方式，此参数只有当数据库类型为DWS时启用，当需要自动创建DWS数据库表，指定表的数据存储方式：</p> <ul style="list-style-type: none"> <li>• ROW：表的数据以行式存储。</li> <li>• COLUMN：表的数据以列式存储。</li> </ul>
toJobConfig.isCompress	否	Boolean	是否压缩，此参数只有当数据库类型为DWS时启用，当需要自动创建DWS数据库表，指定是否对表的数据进行压缩存储。
toJobConfig.useStageTable	否	Boolean	先导入阶段表，如果设置为“true”，数据导入目的表之前会把数据先导入阶段表，如果成功导入阶段表，则再从阶段表导入到目的表，这样避免导入过程失败，在目的表遗留部分成功数据。
toJobConfig.extendCharLength	否	Boolean	扩大字符字段长度，如果设置为“true”，当需要自动创建目的表时，目标表的字符类型字段长度设置为源表相应字段长度的3倍。
toJobConfig.useNullable	否	Boolean	当选择自动创建目的表时，如果选择使用非空约束，则目的表字段的是否非空约束，与原表具有相应非空约束的字段保持一致。

### 4.4.3.2 目的端为 OBS

#### JSON 样例

```
"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.bucketName",
          "value": "cdm"
        },
        {
          "name": "toJobConfig.outputDirectory",
          "value": "/obsfrom/advance/"
        },
        {
          "name": "toJobConfig.outputFormat",
          "value": "CSV_FILE"
        },
        {
          "name": "toJobConfig.fieldSeparator",
          "value": ","
        },
        {
          "name": "toJobConfig.writeToTempFile",
          "value": "false"
        },
        {
          "name": "toJobConfig.validateMD5",
          "value": "false"
        },
        {
          "name": "toJobConfig.recordMD5Result",
          "value": "false"
        },
        {
          "name": "toJobConfig.encodeType",
          "value": "UTF-8"
        },
        {
          "name": "toJobConfig.markerFile",
          "value": "finish.txt"
        },
        {
          "name": "toJobConfig.duplicateFileOpType",
          "value": "REPLACE"
        },
        {
          "name": "toJobConfig.columnList",
          "value": "1&2"
        },
        {
          "name": "toJobConfig.quoteChar",
          "value": "false"
        },
        {
          "name": "toJobConfig.encryption",
          "value": "NONE"
        },
        {
          "name": "toJobConfig.copyContentType",
          "value": "false"
        }
      ],
      "name": "toJobConfig"
    }
  ]
}
```

## 参数说明

参数	是否必选	类型	说明
toJobConfig.bucketName	是	String	OBS的桶名，例如“cdm”。
toJobConfig.outputDirectory	是	String	数据写入路径，例如“data_dir”。
toJobConfig.outputFormat	是	枚举	写入数据时所用的文件格式（二进制除外），支持以下文件格式： <ul style="list-style-type: none"> <li>• CSV_FILE：按照CSV格式写入数据。</li> <li>• BINARY_FILE：二进制格式，不解析文件内容直接传输，CDM会原样写入文件，不改变原始文件格式。</li> </ul> 当选择“BINARY_FILE”时，源端也必须为文件系统。
toJobConfig.fieldSeparator	否	String	列分割符号，当“toJobConfig.outputFormat”（文件格式）为“CSV_FILE”时此参数有效，默认值为：“，”。
toJobConfig.lineSeparator	否	String	行分割符号，当“toJobConfig.outputFormat”（文件格式）为“CSV_FILE”时此参数有效，默认值为：“\r\n”。
toJobConfig.writeFileSize	否	String	源端为数据库时该参数有效，支持按大小分成多个文件存储，避免导出的文件过大，单位为MB。
toJobConfig.duplicateFileType	否	枚举	重复文件处理方式，只有文件名和文件大小都相同才会判定为重复文件。重复文件支持以下处理方式： <ul style="list-style-type: none"> <li>• REPLACE：替换重复文件。</li> <li>• SKIP：跳过重复文件。</li> <li>• ABANDON：发现重复文件停止任务。</li> </ul>
toJobConfig.columnList	否	String	需要抽取的字段列表，字段名之间使用“&”分割，例如：“id&gid&name”。

参数	是否必选	类型	说明
toJobConfig.encrypted	否	枚举	选择是否对上传的数据进行加密，以及加密方式： <ul style="list-style-type: none"> <li>NONE：不加密，直接写入数据。</li> <li>KMS：使用数据加密服务中的KMS进行加密。如果启用KMS加密则无法进行数据的MD5校验。</li> </ul>
toJobConfig.kmsID	否	String	上传时加密使用的密钥。需先在密钥管理服务中创建密钥。
toJobConfig.projectID	否	String	KMS密钥所属的项目ID。
toJobConfig.writeToTempFile	否	Boolean	将二进制文件先写入到临时文件（临时文件以“.tmp”作为后缀），迁移成功后，再进行rename或move操作，在目的端恢复文件。
toJobConfig.validateMD5	否	Boolean	选择是否校验MD5值，不能与KMS加密同时使用。使用二进制格式传输文件时，才能校验MD5值。 计算源文件的MD5值，并与OBS返回的MD5值进行校验。如果源端已经存在MD5文件，则直接读取源端的MD5文件与OBS返回的MD5值进行校验。
toJobConfig.recordMD5Result	否	Boolean	当选择校验MD5值时，这里配置是否记录校验结果。
toJobConfig.recordMD5Link	否	String	可以指定任意一个OBS连接，将MD5校验结果写入该连接的桶。
toJobConfig.recordMD5Bucket	否	String	写入MD5校验结果的OBS桶。
toJobConfig.recordMD5Directory	否	String	写入MD5校验结果的目录。
toJobConfig.encodeType	否	String	编码类型，例如：“UTF_8”或“GBK”。
toJobConfig.markerFile	否	String	当作业执行成功时，会在写入目录下生成一个标识文件，文件名由用户指定，不指定时默认关闭该功能。

参数	是否必选	类型	说明
toJobConfig.outputFormat yContentType	否	Boolean	<p>“toJobConfig.outputFormat”（文件格式）为“BINARY_FILE”，且源端、目的端都为对象存储时，才有该参数。</p> <p>选择“是”后，迁移对象文件时会复制源文件的Content-Type属性，主要用于静态网站的迁移场景。</p> <p>归档存储的桶不支持设置Content-Type属性，所以如果开启了该参数，目的端选择写入的桶时，必须选择非归档存储的桶。</p>
toJobConfig.outputFormat quoteChar	否	Boolean	<p>“toJobConfig.outputFormat”（文件格式）为“CSV_FILE”，才有该参数，用于将数据库的表迁移到文件系统的场景。</p> <p>选择“是”时，如果源端数据表中的某一个字段内容包含字段分隔符或换行符，写入目的端时CDM会使用双引号（"）作为包围符将该字段内容括起来，作为一个整体存储，避免其中的字段分隔符误将一个字段分隔成两个，或者换行符误将字段换行。例如：数据库中某字段为hello.world，使用包围符后，导出到CSV文件的时候数据为"hello.world"。</p>
toJobConfig.outputFormat firstRowAsHeader	否	Boolean	<p>“toJobConfig.outputFormat”（文件格式）为“CSV_FILE”时才有该参数。</p> <p>在迁移表到CSV文件时，CDM默认是不迁移表的标题行，如果该参数选择“是”，CDM在才会将表的标题行数据写入文件。</p>

参数	是否必选	类型	说明
toJobConfig.file Prefix	否	String	自定义文件名前缀，支持任意自定义名称，也支持表名宏时间宏版本宏。例如：\${tableName}_\${dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)}_\${version}。 注意文件名格式要符合obs文件路径命名规范。

### 4.4.3.3 目的端为 HDFS

#### JSON 样例

```
"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.outputDirectory",
          "value": "/hdfsto"
        },
        {
          "name": "toJobConfig.outputFormat",
          "value": "BINARY_FILE"
        },
        {
          "name": "toJobConfig.writeToTempFile",
          "value": "false"
        },
        {
          "name": "toJobConfig.duplicateFileOpType",
          "value": "REPLACE"
        },
        {
          "name": "toJobConfig.compression",
          "value": "NONE"
        },
        {
          "name": "toJobConfig.appendMode",
          "value": "true"
        }
      ],
      "name": "toJobConfig"
    }
  ]
}
```

#### 参数说明

参数	是否必选	类型	说明
toJobConfig.outputDirectory	是	String	数据写入的路径，例如“/data_dir”。

参数	是否必选	类型	说明
toJobConfig.outputFormat	是	枚举	<p>写入数据时所用的文件格式（二进制除外），支持以下文件格式：</p> <ul style="list-style-type: none"> <li>• CSV_FILE：按照CSV格式写入数据。</li> <li>• BINARY_FILE：二进制格式，不解析文件内容直接传输，CDM会原样写入文件，不改变原始文件格式。</li> </ul> <p>当选择“BINARY_FILE”时，源端也必须为文件系统。</p>
toJobConfig.lineSeparator	否	String	<p>行分割符号，当“toJobConfig.outputFormat”（文件格式）为“CSV_FILE”时此参数有效，默认值为：“\r\n”。</p>
toJobConfig.fieldSeparator	否	String	<p>列分割符号，当“toJobConfig.outputFormat”（文件格式）为“CSV_FILE”时此参数有效，默认值为：“，”。</p>
toJobConfig.writeToTempFile	否	Boolean	<p>将二进制文件先写入到临时文件（临时文件以“.tmp”作为后缀），迁移成功后，再进行rename或move操作，在目的端恢复文件。</p>
toJobConfig.duplicateFileType	否	枚举	<p>重复文件处理方式，只有文件名和文件大小都相同才会判定为重复文件。重复文件支持以下处理方式：</p> <ul style="list-style-type: none"> <li>• REPLACE：替换重复文件。</li> <li>• SKIP：跳过重复文件。</li> <li>• ABANDON：发现重复文件停止任务。</li> </ul>



参数	是否必选	类型	说明
toJobConfig.compression	否	枚举	<p>写入文件后，选择对文件的压缩格式。支持以下压缩格式：</p> <ul style="list-style-type: none"> <li>• NONE：不压缩。</li> <li>• DEFLATE：压缩为DEFLATE格式。</li> <li>• GZIP：压缩为GZIP格式。</li> <li>• BZIP2：压缩为BZIP2格式。</li> <li>• LZ4：压缩为LZ4格式。</li> <li>• SNAPPY：压缩为SNAPPY格式。</li> </ul>
toJobConfig.appendMode	是	Boolean	<p>当加载路径已经存在文件，是否需要写入，默认值为“false”。</p>
toJobConfig.encryption	否	枚举	<p>当“toJobConfig.outputFormat”（文件格式）为“BINARY_FILE”（二进制）时才有该参数，选择是否对导入的数据进行加密，以及加密方式：</p> <ul style="list-style-type: none"> <li>• NONE：不加密，直接写入数据。</li> <li>• AES-256-GCM：使用长度为256byte的AES对称加密算法，目前加密算法只支持AES-256-GCM（NoPadding）。</li> </ul>
toJobConfig.dek	否	String	<p>数据加密密钥，“toJobConfig.encryption”（加密方式）选择“AES-256-GCM”时有该参数，密钥由长度64的十六进制数组成。</p> <p>请您牢记这里配置的密钥，解密时的密钥与这里配置的必须一致。如果不一致系统不会报异常，只是解密出来的数据会错误。</p>

参数	是否必选	类型	说明
toJobConfig.iv	否	String	初始化向量， “toJobConfig.encryption”（加密方式）选择“AES-256-GCM”时有该参数，初始化向量由长度32的十六进制数组组成。  请您牢记这里配置的初始化向量，解密时的初始化向量与这里配置的必须一致。如果不一致系统不会报异常，只是解密出来的数据会错误。
toJobConfig.file Prefix	否	String	自定义文件名前缀，支持时间宏。例如：test_\${dateformat(yyyyMMdd, -1, DAY)}  注意文件名格式要符合hdfs文件路径命名规范。

#### 4.4.3.4 目的端为 Hive

##### JSON 样例

```
"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.hive",
          "value": "hive"
        },
        {
          "name": "toJobConfig.database",
          "value": "rf_database"
        },
        {
          "name": "toJobConfig.table",
          "value": "rf_to"
        },
        {
          "name": "toJobConfig.tablePreparation",
          "value": "DO_NOTHING"
        },
        {
          "name": "toJobConfig.columnList",
          "value": "aa&bb&cc&dd"
        },
        {
          "name": "toJobConfig.shouldClearTable",
          "value": "true"
        }
      ],
      "name": "toJobConfig"
    }
  ]
}
```

## 参数说明

参数	是否必选	类型	说明
toJobConfig.hive	否	String	写入数据的数据源。
toJobConfig.database	否	String	写入数据的数据库名称，例如：“default”。
toJobConfig.table	是	String	写入数据的表名。
toJobConfig.tablePreparation	是	枚举	写入表数据时，用户选择的操作： <ul style="list-style-type: none"> <li>• DO_NOTHING：不自动建表。</li> <li>• CREATE_WHEN_NOT_EXISTS：当目的端的数据库没有“tableName”参数中指定的表时，CDM会自动创建该表。</li> <li>• DROP_AND_CREATE：先删除“tableName”参数中指定的表，然后再重新创建该表。</li> </ul>
toJobConfig.columnList	否	String	需要加载的字段列表，字段名之间使用“&”分割，例如：“id&gid&name”。
toJobConfig.shouldClearTable	否	Boolean	导入前是否清空目标表的数据，如果设置为true，任务启动前会清除目标表中数据。

### 4.4.3.5 目的端为 HBase/CloudTable

#### JSON 样例

```

"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.table",
          "value": "rf_to"
        },
        {
          "name": "toJobConfig.storageType",
          "value": "PUTLIST"
        },
        {
          "name": "toJobConfig.columns",
          "value": "AA:AA&BB:BB&CC:CC&DD:DD"
        }
      ],
      {

```

```

    "name": "toJobConfig.rowKeyColumn",
    "value": "AA:AA"
  },
  {
    "name": "toJobConfig.isOverride",
    "value": "false"
  },
  {
    "name": "toJobConfig.isRowkeyRedundancy",
    "value": "false"
  },
  {
    "name": "toJobConfig.algorithm",
    "value": "NONE"
  },
  {
    "name": "toJobConfig.writeToWAL",
    "value": "true"
  },
  {
    "name": "toJobConfig.transType",
    "value": "false"
  }
],
"name": "toJobConfig"
}
]
}

```

## 参数说明

参数	是否必选	类型	说明
toJobConfig.table	是	String	写入数据的表名，例如：“TBL_EXAMPLE”。
toJobConfig.storageType	是	枚举	将数据写入到HBase表中的方法： <ul style="list-style-type: none"> <li>● PUTLIST：put list方式写入。</li> </ul>
toJobConfig.columns	否	String	需要抽取数据的列，列号之间使用“&”分割，列族与列之间用“:”分隔，例如：“cf1:c1&cf2:c2”。
toJobConfig.rowKeyColumn	是	String	作为rowkey的列，列号之间使用“&”分割，列族与列之间用“:”分隔，例如：“cf1:c1&cf2:c2”。
toJobConfig.isOverride	否	Boolean	使用BULKLOAD方式导入数据时，是否清空数据，例如：“true”。
toJobConfig.delimiter	否	String	当选取多个列做rowkey时，连接多列的分隔符，例如：“ ”。

参数	是否必选	类型	说明
toJobConfig.isRowkeyRedundancy	否	Boolean	是否将选做Rowkey的数据同时写入HBase的列。
toJobConfig.algorithm	否	枚举	创建新HBase表时采用的压缩算法，支持SNAPPY和GZ算法，默认为“NONE”。
toJobConfig.writeToWAL	否	Boolean	选择是否开启HBase的预写日志机制（WAL，Write Ahead Log）。 <ul style="list-style-type: none"> <li>是：开启后如果出现HBase服务器宕机，则可以从WAL中回放执行之前没有完成的操作。</li> <li>否：关闭时能提升写入性能，但如果HBase服务器宕机可能会造成数据丢失。</li> </ul>
toJobConfig.transType	否	Boolean	<ul style="list-style-type: none"> <li>true：源端数据库中的Short、Int、Long、Float、Double、Decimal类型列的数据，会转换为Byte[]数组（二进制）写入HBase，其他类型的按字符串写入。如果这几种类型中，有合并做rowkey的，就依然当字符串写入。该功能作用是：降低存储占用空间，存储更高效；特定场景下rowkey分布更均匀。</li> <li>false：源端数据库中所有类型的数据，都会按照字符串写入HBase。</li> </ul>

#### 4.4.3.6 目的端为 DDS

##### JSON 样例

```
"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.database",
          "value": "demo"
        },
        {
          "name": "toJobConfig.collectionName",
          "value": "cdmbase"
        }
      ]
    }
  ]
}
```

```

        "name": "toJobConfig.columnList",
        "value": "_char&_varchar"
    },
    {
        "name": "toJobConfig.isBatchMigration",
        "value": "false"
    }
  ],
  "name": "toJobConfig"
}
]
}

```

## 参数说明

参数	是否必选	类型	说明
toJobConfig.database	是	String	MongoDB/DDS的数据库名称
toJobConfig.collectionName	是	String	MongoDB/DDS的集合名称。
toJobConfig.columnList	否	String	需要抽取的字段列表，字段名之间使用“&”分割，例如：“id&gid&name”。
toJobConfig.isBatchMigration	否	Boolean	是否为整库迁移。

### 4.4.3.7 目的端为 Elasticsearch/云搜索服务

#### JSON 样例

```

"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.index",
          "value": "cdm"
        },
        {
          "name": "toJobConfig.type",
          "value": "type1"
        },
        {
          "name": "toJobConfig.shouldClearType",
          "value": "false"
        },
        {
          "name": "toJobConfig.pipeLine",
          "value": "es_03"
        }
      ],
      "name": "toJobConfig"
    }
  ]
}

```

## 参数说明

参数	是否必选	类型	说明
toJobConfig.index	是	String	写入数据的索引，类似关系数据库中的数据库名称。
toJobConfig.type	是	String	写入数据的类型，类似关系数据库中的表名。
toJobConfig.shouldClearType	否	Boolean	导入前是否清除数据。
toJobConfig.primaryKey	否	String	主键或唯一索引。
toJobConfig.columnList	否	String	需要写入的字段列表，字段名之间使用“&”分隔，例如：“id&gid&name”。
toJobConfig.pipeline	否	String	需要先在kibana中创建管道ID，这里才可以选择，该参数用于数据传到云搜索服务/Elasticsearch后，通过Elasticsearch的数据转换pipeline进行数据格式变换。

参数	是否必选	类型	说明
toJobConfig.createIndexStrategy	否	枚举	<p>对于持续写入数据到Elasticsearch的流式作业，CDM支持在Elasticsearch中定时创建新索引并写入数据，方便用户后期删除过期的数据。支持按以下周期创建新索引：</p> <ul style="list-style-type: none"> <li>• EveryHour：每小时整点创建新索引，新索引的命名格式为“索引名+年+月+日+小时”，例如“index2018121709”。</li> <li>• EveryDay：每天零点零分创建新索引，新索引的命名格式为“索引名+年+月+日”，例如“index20181217”。</li> <li>• EveryWeek：每周周一的零点零分创建新索引，新索引的命名格式为“索引名+年+周”，例如“index201842”。</li> <li>• EveryMonth：每月一号零点零分创建新索引，新索引的命名格式为“索引名+年+月”，例如“index201812”。</li> </ul> <p>从文件类抽取数据时，必须配置单个抽取（“抽取并发数”参数配置为1），否则该参数无效。</p>

#### 4.4.3.8 目的端为 DLI

##### JSON 样例

```

"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.queue",
          "value": "cdm"
        },
        {
          "name": "toJobConfig.database",
          "value": "sqoop"
        },
        {
          "name": "toJobConfig.table",
          "value": "est1"
        },
        {
          "name": "toJobConfig.columnList",
          "value":
    
```



```
"string_&int_&date_&double_&boolean_&short_&timestamp_&long_&smallint_&bigint_"
    },
    {
      "name": "toJobConfig.shouldClearTable",
      "value": "false"
    }
  ],
  "name": "toJobConfig"
}
]
```

## 参数说明

参数	是否必选	类型	说明
toJobConfig.queue	是	String	写入数据的资源队列。
toJobConfig.database	是	String	写入数据到数据湖探索（DLI）的哪个数据库。
toJobConfig.table	是	String	写入数据的表名。
toJobConfig.columnList	否	String	需要加载的字段列表，字段名之间使用“&”分割，例如：“id&gid&name”。
toJobConfig.shouldClearTable	否	Boolean	导入前是否清空资源队列的数据。

### 4.4.3.9 目的端为 DIS

#### JSON 样例

```
"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.streamName",
          "value": "cdm"
        },
        {
          "name": "toJobConfig.separator",
          "value": ","
        },
        {
          "name": "toJobConfig.identifierEnclose",
          "value": ""
        }
      ],
      "name": "toJobConfig"
    }
  ]
}
```

## 参数说明

参数	是否必选	类型	说明
toJobConfig.streamName	是	String	DIS的通道名。
toJobConfig.separator	否	String	字段分隔符，默认为空格。
toJobConfig.identifierEnclose	否	String	连接引用表名或列名时的分隔符号，默认为空。

### 4.4.4 作业任务参数说明

在[指定集群创建作业](#)或者[随机集群创建作业并执行](#)时，由“driver-config-values”参数指定作业任务配置，包含如下功能：

- 作业失败重试：如果作业执行失败，可选择是否自动重新启动作业。
- 作业分组：CDM支持对作业进行分组，分组后的作业可以按组显示、按组批量删除作业、按组批量启动作业、按组导出等。
- 是否定时执行：可选择作业是否定时自动启动。
- 抽取并发数：可设置同时执行的抽取任务数。
- 是否写入脏数据：如果需要将作业执行过程中处理失败的数据、或者被清洗过滤掉的数据写入OBS中，以便后面查看，可通过该参数配置，写入脏数据前需要先配置好OBS连接。
- 作业运行完是否删除：可选择是否自动删除作业。

### JSON 样例

```
"driver-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "throttlingConfig.numExtractors",
          "value": "1"
        },
        {
          "name": "throttlingConfig.numLoaders",
          "value": "1"
        },
        {
          "name": "throttlingConfig.recordDirtyData",
          "value": "false"
        }
      ],
      "name": "throttlingConfig"
    },
    {
      "inputs": [],
      "name": "jarConfig"
    },
    {
      "inputs": [
        {
          "name": "schedulerConfig.isSchedulerJob",
```

```

        "value": "false"
      },
      {
        "name": "schedulerConfig.disposableType",
        "value": "NONE"
      }
    ],
    "name": "schedulerConfig"
  },
  {
    "inputs": [],
    "name": "transformConfig"
  },
  {
    "inputs": [
      {
        "name": "retryJobConfig.retryJobType",
        "value": "NONE"
      }
    ],
    "name": "retryJobConfig"
  }
]
}

```

## 参数说明

参数	是否必选	类型	说明
throttlingConfig.numExtractors	否	Integer	最大抽取任务并发数，例如：“20”。
groupJobConfig.groupName	否	枚举	选择作业所属的分组，默认分组为“DEFAULT”。
throttlingConfig.numLoaders	否	Integer	仅当HBase或Hive作为目的数据源时该参数才有效。 最大加载任务数，例如：“5”。
throttlingConfig.recordDirtyData	否	Boolean	是否写入脏数据，例如：“true”。
throttlingConfig.writeToLink	否	String	脏数据要写入的连接，目前只支持写入到OBS连接或HDFS连接。例如：“obslink”。
throttlingConfig.obsBucket	否	String	写入脏数据的OBS桶的名称，只有当脏数据要写入OBS连接的时候，此参数才生效。例如：“dirtyData”。
throttlingConfig.dirtyDataDirectory	否	String	写入脏数据的目录： <ul style="list-style-type: none"> <li>如果选择写入到HDFS，此参数即为HDFS目录。</li> <li>如果选择写入到OBS，此参数表示相应的OBS桶下的目录，例如：“/data/dirtydata/”。</li> </ul>

参数	是否必选	类型	说明
throttlingConfig.maxErrorRecords	否	String	单个分片的最大错误记录数。单个map的错误记录超过设置的最大错误记录数时，任务自动结束，已经导入的数据不回退。
schedulerConfig.isSchedulerJob	否	Boolean	是否开启定时任务，例如：“true”。
schedulerConfig.cycleType	否	String	定时任务的周期类型，目前支持五种周期类型： <ul style="list-style-type: none"> <li>• minute: 分钟</li> <li>• hour: 小时</li> <li>• day: 天</li> <li>• week: 周</li> <li>• month: 月</li> </ul>
schedulerConfig.cycle	否	Integer	定时任务的周期，如果周期类型选择了“minute”，“cycle”输入“10”，就表示该定时任务每10分钟执行一次。
schedulerConfig.runAt	否	String	定时任务在周期内的触发时间，当周期为“hour”、“week”或“month”时，该参数有效。 <ul style="list-style-type: none"> <li>• 如果周期类型为“month”，设定周期为“1”，“runAt”输入“15”，就表示每个月的15号执行该定时任务。并且该参数支持输入多个，以英文的逗号“,”分隔开。例如上述场景中“runAt”输入“1,2,3,4,5”，就表示每个月的1日、2日、3日、4日和5日执行该定时任务。</li> <li>• 周期类型为“week”时，“runAt”输入“mon,tue,wed,thu,fri”，就表示周一到周五执行该定时任务。</li> <li>• 周期类型为“hour”时，“runAt”输入“27,57”，表示周期内的27分和57分执行该定时任务。</li> </ul>
schedulerConfig.startDate	否	String	定时任务的开始时间，例如：“2018-01-24 19:56:19”。

参数	是否必选	类型	说明
schedulerConfig. stopDate	否	String	定时任务的结束日期，例如： “2018-01-27 23:59:00”。 如果不输入结束时间，则表示定 时任务将一直执行，永远不会结 束。
schedulerConfig. disposableType	否	枚举	作业运行完是否删除： <ul style="list-style-type: none"> <li>• NONE：作业执行完不删除。</li> <li>• DELETE_AFTER_SUCCEED： 仅作业执行成功时删除该作 业，适合海量一次性作业。</li> <li>• DELETE：作业执行完删除该 作业，执行成功或失败都会删 除。</li> </ul>
retryJobConfig.r etryJobType	否	枚举	如果作业执行失败，选择是否自 动重试： <ul style="list-style-type: none"> <li>• NONE：不重试。</li> <li>• RETRY_TRIPLE：重试三次。</li> </ul>

# 5 数据开发 API (V1)

## 5.1 脚本开发 API

### 5.1.1 创建脚本

#### 功能介绍

创建一个新脚本，当前支持的脚本类型有DLI SQL、Flink SQL、RDS SQL、Spark SQL、Hive SQL、DWS SQL、Shell、Presto SQL、ClickHouse SQL、HetuEngine SQL、Python、Spark Python和Impala SQL。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
POST /v1/{project\_id}/scripts
- 参数说明

表 5-1 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 5-2 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

表 5-3 Script 参数说明

参数名	是否必选	参数类型	说明
name	是	String	脚本名称，只能包含五种字符：英文字母、数字、中文、中划线和点号，且长度小于等于128个字符。脚本名称不能重复。
type	是	String	脚本类型，包含： <ul style="list-style-type: none"> <li>• FlinkSQL</li> <li>• DLISQL</li> <li>• SparkSQL</li> <li>• HiveSQL</li> <li>• DWSSQL</li> <li>• RDSSQL</li> <li>• Shell</li> <li>• PRESTO</li> <li>• ClickHouseSQL</li> <li>• HetuEngineSQL</li> <li>• PYTHON</li> <li>• ImpalaSQL</li> <li>• SparkPython</li> </ul>
content	是	String	脚本内容，最大支持4M。
directory	否	String	脚本所在目录路径。 通过DataArts Studio管理控制台 > 数据开发，左侧列表选择“数据开发 > 脚本开发”。在脚本的目录树上，可以查看到当前已经创建的目录，默认在根目录/。

参数名	是否必选	参数类型	说明
connectionName	否	String	脚本关联的连接名称。当type参数值为DLISQL、SparkSQL、HiveSQL、DWSSQL、Shell、PRESTO、ClickHouseSQL、HetuEngineSQL、RDSSQL、ImpalaSQL、PYTHON、SparkPython其中之一时，这个参数是必选的。用户可以通过 <a href="#">查询连接列表（待下线）</a> 接口获取当前系统中已经存在的连接。默认值为空。
database	否	String	执行SQL语句所关联的数据库，当type参数值为DLISQL、SparkSQL、HiveSQL、DWSSQL、PRESTO、ClickHouseSQL、ImpalaSQL、HetuEngineSQL、RDSSQL其中之一时，才支持此参数。 <ul style="list-style-type: none"> <li>type为DLI SQL时，可以通过<a href="#">查看所有数据库</a>接口获取数据库信息。</li> <li>type为其他类型的时候，必选。</li> </ul>
queueName	是	String	DLI资源队列名称，当type参数值为DLI SQL时，才支持此参数。可以通过 <a href="#">查询队列列表接口</a> 获取队列信息。默认值为空。
configuration	否	map<String, Object>	用户定义适用于此作业的配置参数，当type参数值为DLI SQL时，才支持此参数。当前支持的配置项列表请参考 <a href="#">conf参数说明</a> 。默认值为空。
description	否	String	描述，长度不能超过255个字符。
owner	否	String	责任人名称
targetStatus	否	String	在开启审批开关后，需要填写该字段。表示创建脚本的目标状态，有三种状态：SAVED、SUBMITTED和PRODUCTION，分别表示脚本创建后是保存态，提交态，生产态。 <ul style="list-style-type: none"> <li>保存态表示脚本仅保存，无法调度运行，需要提交并审核通过后才能运行。</li> <li>提交态表示脚本保存后会自动提交，需要审核通过才能运行。</li> <li>生产态表示脚本跳过审批环节，创建后可以直接运行。注意：只有工作空间的管理员用户才能创建生产态的脚本。审批开关的配置请参见<a href="#">审批中心</a>。</li> </ul>
approvers	否	List<JobApprover>	在开启审批开关后，需要填写该字段，表示脚本审批人，参考 <a href="#">表5-4</a> 。



表 5-4 审批人属性说明

参数名	是否必选	参数类型	说明
approverName	是	String	审批人名称。

## 响应参数

无

## 请求示例

创建一个脚本名称为echoTimeShell的脚本，脚本类型为Shell，脚本内容为echo a，脚本关联的连接名称为con。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts
{
  "name": "echoTimeShell",
  "type": "Shell",
  "content": "echo a",
  "queueName": "default",
  "connectionName": "con"
}
```

开启审批开关情况下，创建一个脚本。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts
{
  "name": "echoTimeShell",
  "type": "Shell",
  "content": "echo a",
  "queueName": "default",
  "connectionName": "con",
  "targetStatus": "SUBMITTED",
  "approvers": [
    {
      "approverName": "userName1"
    },
    {
      "approverName": "userName2"
    }
  ]
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.6247",
  "error_msg": "The script type is not specified."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.1.2 修改脚本内容

### 功能介绍

本接口用以修改脚本的配置项或者脚本内容。

- 修改脚本时，需要指定待修改的脚本名称。
- 脚本名称和脚本类型不支持修改。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
PUT /v1/{project\_id}/scripts/{script\_name}
- 参数说明

表 5-5 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
script_name	是	String	脚本名称。

### 请求参数

表 5-6 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"><li>• 如果不设置该参数，默认查询default工作空间下的数据。</li><li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li></ul>

表 5-7 Script 参数说明

参数名	是否必选	参数类型	说明
name	是	String	脚本名称，只能包含五种字符：英文字母、数字、中文、中划线和点号，且长度小于等于128个字符。脚本名称不能重复。
type	是	String	脚本类型，包含： <ul style="list-style-type: none"> <li>• FlinkSQL</li> <li>• DLISQL</li> <li>• SparkSQL</li> <li>• HiveSQL</li> <li>• DWSSQL</li> <li>• RDSSQL</li> <li>• Shell</li> <li>• PRESTO</li> <li>• ClickHouseSQL</li> <li>• HetuEngineSQL</li> <li>• PYTHON</li> <li>• ImpalaSQL</li> <li>• SparkPython</li> </ul>
content	是	String	脚本内容，最大支持4M。
directory	否	String	脚本所在目录路径。 通过DataArts Studio管理控制台 > 数据开发，左侧列表选择“数据开发 > 脚本开发”。在脚本的目录树上，可以查看到当前已经创建的目录，默认在根目录/。
connectionName	否	String	脚本关联的连接名称。当type参数值为DLISQL、SparkSQL、HiveSQL、DWSSQL、Shell、PRESTO、ClickHouseSQL、HetuEngineSQL、RDSSQL、ImpalaSQL、PYTHON、SparkPython其中之一时，这个参数是必选的。用户可以通过 <a href="#">查询连接列表（待下线）</a> 接口获取当前系统中已经存在的连接。默认值为空。

参数名	是否必选	参数类型	说明
database	否	String	<p>执行SQL语句所关联的数据库，当type参数值为DLISQL、SparkSQL、HiveSQL、DWSSQL、PRESTO、ClickHouseSQL、ImpalaSQL、HetuEngineSQL、RDSSQL其中之一时，才支持此参数。</p> <ul style="list-style-type: none"> <li>type为DLI SQL时，可以通过<a href="#">查看所有数据库</a>接口获取数据库信息。</li> <li>type为其他类型的时候，需要通过JDBC方式连上集群，查询数据库信息。默认值为空。</li> </ul>
queueName	是	String	<p>DLI资源队列名称，当type参数值为DLISQL时，才支持此参数。可以通过<a href="#">查询队列列表接口</a>获取队列信息。默认值为空。</p>
configuration	否	map<String, Object>	<p>用户定义适用于此作业的配置参数，当type参数值为DLISQL时，才支持此参数。当前支持的配置项列表请参考 <a href="#">conf参数说明</a>。默认值为空。</p>
description	否	String	<p>描述，长度不能超过255个字符。</p>
targetStatus	否	String	<p>在开启审批开关后，需要填写该字段。表示创建脚本的目标状态，有三种状态：SAVED、SUBMITTED和PRODUCTION，分别表示脚本创建后是保存态，提交态，生产态。</p> <ul style="list-style-type: none"> <li>保存态表示脚本仅保存，无法调度运行，需要提交并审核通过后才能运行。</li> <li>提交态表示脚本保存后会自动提交，需要审核通过才能运行。</li> <li>生产态表示脚本跳过审批环节，创建后可以直接运行。注意：只有工作空间的管理员用户才能创建生产态的脚本。审批开关的配置请参见<a href="#">审批中心</a>。</li> </ul>
approvers	否	List<JobApprover>	<p>在开启审批开关后，需要填写该字段，表示脚本审批人，参考<a href="#">表5-8</a>。审批开关的配置请参见<a href="#">审批中心</a>。</p>

表 5-8 审批人属性说明

参数名	是否必选	参数类型	说明
approverName	是	String	审批人名称。

## 响应参数

无

## 请求示例

修改脚本名称为echoTimeShell的脚本内容，脚本类型为Shell，脚本内容为echo a，脚本关联的连接名称为con。

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/echoTimeShell
{
  "name": "echoTimeShell",
  "type": "Shell",
  "content": "echo a",
  "queueName": "default",
  "connectionName": "con"
}
```

开启审批开关的情况下，修改脚本内容。

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/echoTimeShell
{
  "name": "echoTimeShell",
  "type": "Shell",
  "content": "echo a",
  "queueName": "default",
  "connectionName": "con",
  "approvers": [
    {
      "approverName": "userName1"
    },
    {
      "approverName": "userName2"
    }
  ]
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.6201",
  "error_msg": "The script does not exist."
}
```

## 5.1.3 查询脚本信息

### 功能介绍

此接口可以查询指定脚本的定义，包括脚本类型和脚本内容。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式

GET /v1/{project\_id}/scripts/{script\_name}?version={version}

- 参数说明

表 5-9 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
script_name	是	String	脚本名称。
version	否	integer	脚本版本号，若传入版本号，则查询指定版本号的脚本；若不传入，则查询最新的版本脚本。
status	否	String	当不指定version时，可通过status查询脚本的状态。默认为SAVED，即查询的是最新提交的版本，当status为PUBLISHED时查询的是脚本的生产版本。

## 请求参数

表 5-10 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

表 5-11 Script 参数说明

参数名	是否必选	参数类型	说明
name	是	String	脚本名称，只能包含五种字符：英文字母、数字、中文、中划线和点号，且长度小于等于128个字符。脚本名称不能重复。
id	是	String	脚本ID。

参数名	是否必选	参数类型	说明
type	是	String	脚本类型，包含： <ul style="list-style-type: none"> <li>• FlinkSQL</li> <li>• DLISQL</li> <li>• SparkSQL</li> <li>• HiveSQL</li> <li>• DWSSQL</li> <li>• RDSSQL</li> <li>• Shell</li> <li>• PRESTO</li> <li>• ClickHouseSQL</li> <li>• HetuEngineSQL</li> <li>• PYTHON</li> <li>• ImpalaSQL</li> <li>• SparkPython</li> </ul>
content	是	String	脚本内容，最大支持64KB。
directory	否	String	脚本所在目录路径。 通过DataArts Studio管理控制台 > 数据开发，左侧列表选择“数据开发 > 脚本开发”。在脚本的目录树上，可以查看到当前已经创建的目录，默认在根目录/。
connectionName	否	String	脚本关联的连接名称。当type参数值为DLISQL、SparkSQL、HiveSQL、DWSSQL、Shell、PRESTO、ClickHouseSQL、ImpalaSQL、HetuEngineSQL、RDSSQL其中之一时，这个参数是必选的。用户可以通过 <a href="#">查询连接列表（待下线）</a> 接口获取当前系统中已经存在的连接。默认值为空。
database	否	String	执行SQL语句所关联的数据库，当type参数值为DLISQL、SparkSQL、HiveSQL、DWSSQL、PRESTO、ClickHouseSQL、ImpalaSQL、HetuEngineSQL、RDSSQL其中之一时，才支持此参数。 <ul style="list-style-type: none"> <li>• type为DLISQL时，可以通过<a href="#">查看所有数据库</a>接口获取数据库信息。</li> <li>• type为其他类型的时候，需要通过JDBC方式连上集群，查询数据库信息。默认值为空。</li> </ul>

参数名	是否必选	参数类型	说明
queueName	否	String	DLI资源队列名称，当type参数值为DLISQL时，才支持此参数。可以通过 <a href="#">查询队列列表接口</a> 获取队列信息。默认值为空。
configuration	否	map <String, Object>	用户定义适用于此作业的配置参数，当type参数值为DLISQL时，才支持此参数。当前支持的配置项列表请参考 <a href="#">conf参数说明</a> 。默认值为空。
description	否	String	描述，长度不能超过255个字符。
autoAcquireLock	否	boolean	自动获取编辑锁参数，默认为true。

## 请求示例

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/scriptName
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "configuration": {},
  "connectionName": "mrs_spark",
  "content": "SELECT 1;",
  "database": "aaa",
  "description": "",
  "directory": "/",
  "id": "8abfd8a49035cf8a019035d37b550000",
  "name": "mrs_spark_sql",
  "type": "SparkSQL"
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF6201",
  "error_msg": "The script does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.1.4 查询脚本列表

### 功能介绍

此接口用于查询脚本列表，每次查询返回的脚本数量上限是1000个。



## 调试

您可以在 [API Explorer](#) 中调试该接口，支持自动认证鉴权。

## URI

- URI格式  
GET /v1/{project\_id}/scripts?  
offset={offset}&limit={limit}&scriptName={scriptName}
- 参数说明

表 5-12 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
limit	否	Integer	分页返回结果，指定每页最大记录数，范围[1,100]。 默认值：10。
offset	否	Integer	分页的起始页，默认值为0。取值范围大于等于0。
scriptName	否	String	脚本名称。

## 请求参数

表 5-13 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

参数名	是否必选	参数类型	说明
total	是	Integer	总的脚本个数。
scripts	是	List<Script>	脚本列表。

表 5-14 Script 参数说明

参数名	是否必选	参数类型	说明
name	是	String	脚本名称，只能包含五种字符：英文字母、数字、中文、中划线和点号，且长度小于等于128个字符。脚本名称不能重复。
id	是	String	脚本ID。
type	是	String	脚本类型，包含： <ul style="list-style-type: none"> <li>• FlinkSQL</li> <li>• DLISQL</li> <li>• SparkSQL</li> <li>• HiveSQL</li> <li>• DWSSQL</li> <li>• RDSSQL</li> <li>• Shell</li> <li>• PRESTO</li> <li>• ClickHouseSQL</li> <li>• HetuEngineSQL</li> <li>• PYTHON</li> <li>• ImpalaSQL</li> <li>• SparkPython</li> </ul>
content	是	String	脚本内容，最大支持64KB。
directory	否	String	脚本所在目录路径。 通过DataArts Studio管理控制台 > 数据开发，左侧列表选择“数据开发 > 脚本开发”。在脚本的目录树上，可以查看到当前已经创建的目录，默认在根目录/。
connectionName	否	String	脚本关联的连接名称。当type参数值为DLISQL、SparkSQL、HiveSQL、DWSSQL、Shell、PRESTO、ClickHouseSQL、ImpalaSQL、HetuEngineSQL、RDSSQL其中之一时，这个参数是必选的。用户可以通过 <a href="#">查询连接列表（待下线）</a> 接口获取当前系统中已经存在的连接。默认值为空。

参数名	是否必选	参数类型	说明
database	否	String	<p>执行SQL语句所关联的数据库，当type参数值为DLISQL、SparkSQL、HiveSQL、DWSSQL、PRESTO、ClickHouseSQL、ImpalaSQL、HetuEngineSQL、RDSSQL其中之一时，才支持此参数。</p> <ul style="list-style-type: none"> <li>type为DLISQL时，可以通过<a href="#">查看所有数据库接口</a>获取数据库信息。</li> <li>type为其他类型的时候，需要通过JDBC方式连上集群，查询数据库信息。默认值为空。</li> </ul>
queueName	否	String	<p>DLI资源队列名称，当type参数值为DLISQL时，才支持此参数。可以通过<a href="#">查询队列列表接口</a>获取队列信息。默认值为空。</p>
configuration	否	map <String, Object>	<p>用户定义适用于此作业的配置参数，当type参数值为DLISQL时，才支持此参数。当前支持的配置项列表请参考 <a href="#">conf参数说明</a>。默认值为空。</p>
description	否	String	<p>描述，长度不能超过255个字符。</p>

## 请求示例

查询脚本列表。

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "total": 1,
  "scripts": [
    {
      "configuration": {},
      "connectionName": "mrs_spark",
      "content": "SELECT 1;",
      "database": "aaa",
      "description": "",
      "directory": "/",
      "id": "8abfd8a49035cf8a019035d37b550000",
      "name": "mrs_spark_sql",
      "type": "SparkSQL"
    }
  ]
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.3051",
}
```

```
"error_msg":"The request parameter is invalid."
}
```

## 5.1.5 查询脚本实例执行结果

### 功能介绍

该接口用以获取脚本实例的执行状态和结果。

查询时，需要指定脚本名称和脚本实例ID。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
GET /v1/{project\_id}/scripts/{script\_name}/instances/{instance\_id}
- 参数说明

表 5-15 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
script_name	是	String	脚本名称。
instance_id	是	String	脚本实例id，获取方法请参见 <a href="#">执行脚本的响应参数</a> 。

### 请求参数

表 5-16 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

表 5-17 参数说明

参数名	是否必选	参数类型	说明
status	是	String	执行状态。 LAUNCHING: 提交中 RUNNING: 运行中 FINISHED: 执行成功 FAILED: 执行失败
results	是	List<Result>	执行结果
message	否	String	执行失败消息。

表 5-18 Result 数据结构说明

参数名	是否必选	参数类型	说明
message	否	Object	结构返回信息。
schema	否	List<Map<String, String>>	元数据信息。
rows	否	List<List<Object>>	返回结果数据。
rowCount	否	Long	结果行数。
inputRowCount	否	Long	输入结果的行数。(dli等脚本执行会执行此结果)
resultCount	否	Long	结果行数。(dli等脚本执行会执行此结果)
duration	否	Float	脚本实例的执行时长, 单位是秒。
rawResult	否	Map<String, Object>	脚本结果信息。

## 请求示例

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/dwsscript/instances/a1ad-448a-9d56-4154193d49c5
```

## 响应示例

- 成功响应  
HTTP状态码 200

```
{
  "results": [{
    "message": "",
    "duration": 0.5,
    "rowCount": 1,
    "rows": [[913460.0,
      765.0,
      "8/31/2015 23:26",
      "Harry Bridges Plaza (Ferry Building)",
      50.0,
      "8/31/2015 23:39",
      "San Francisco Caltrain (Townsend at 4th)",
      70.0,
      "288",
      "Subscriber",
      "2139"]],
    "schema": [{
      "TripID": "int"
    },
    {
      "Duration": "int"
    },
    {
      "StartDate": "string"
    },
    {
      "StartStation": "string"
    },
    {
      "StartTerminal": "int"
    },
    {
      "EndDate": "string"
    },
    {
      "EndStation": "string"
    },
    {
      "EndTerminal": "int"
    },
    {
      "Bike": "string"
    },
    {
      "SubscriberType": "string"
    },
    {
      "ZipCode": "string"
    }
  ]
}],
  "status": "FINISHED"
}
```

- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.6201",
  "error_msg": "The script does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.1.6 删除脚本

### 功能介绍

此接口用以删除指定脚本。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
DELETE /v1/{project\_id}/scripts/{script\_name}
- 参数说明

表 5-19 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
script_name	是	String	脚本名称。

### 请求参数

表 5-20 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-21 参数说明

参数名	是否必选	参数类型	说明
approvers	否	List<JobApprover>	在开启审批开关后，需要填写该字段，表示脚本审批人，参考 <a href="#">表5-22</a> 。 审批开关的配置请参见 <a href="#">审批中心</a> 。

表 5-22 审批人属性说明

参数名	是否必选	参数类型	说明
approverName	是	String	审批人名称。

## 响应参数

无

## 请求示例

```
DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/echoTime
```

开启审批开关的情况下，删除脚本。

```
DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/echoTime
{
  "approvers": [
    {
      "approverName": "userName1"
    },
    {
      "approverName": "userName2"
    }
  ]
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.6201",
  "error_msg": "The script does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.1.7 执行脚本

### 功能介绍

该接口用以执行指定脚本，支持执行DWS SQL、DLI SQL、RDS SQL、Hive SQL、Spark SQL、Presto SQL、Shell、ClickHouse SQL、HetuEngine SQL、Python、Spark Python和Impala SQL脚本。脚本每执行一次，生成一个脚本实例。执行脚本接口是一个异步命令，需要通过[查询脚本实例执行结果](#)获取执行状态和结果。

#### 📖 说明

DLI SQL和Presto SQL不支持多语句场景。



## 调试

您可以在 [API Explorer](#) 中调试该接口，支持自动认证鉴权。

## URI

- URI 格式  
POST /v1/{project\_id}/scripts/{script\_name}/execute
- 参数说明

表 5-23 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
script_name	是	String	脚本名称。

## 请求参数

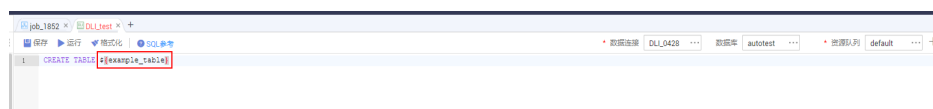
表 5-24 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-25 参数说明

参数名	是否必选	参数类型	说明
params	否	Object	脚本参数Map<String,String>。如果脚本中定义了参数，通过params携带参数值。

脚本参数指的是脚本内容里的参数，如下所示：



## 响应参数

表 5-26 参数说明

参数名	是否必选	参数类型	说明
instanceId	是	String	执行脚本的实例ID。在 <a href="#">查询脚本实例执行结果</a> 使用instanceId获取执行结果。

## 请求示例

执行脚本名称为dws\_sql的脚本，脚本中参数tableVar的值为citys，脚本中参数time的值为2019-07-25。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/dws_sql/execute
{
  "params": {
    "tableVar": "citys",
    "time": "2019-07-25"
  }
}
```

## 响应示例

- 成功响应  
HTTP状态码 200
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF6201",
  "error_msg": "The script does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.1.8 停止脚本实例的执行

### 功能介绍

该接口用以停止脚本实例的执行。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v1/{project\_id}/scripts/{script\_name}/instances/{instance\_id}/stop

- 参数说明

表 5-27 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
script_name	是	String	脚本名称。
instance_id	是	String	脚本实例id，获取方法请参见 <a href="#">执行脚本的响应参数</a> 。

## 请求参数

表 5-28 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

无

## 请求示例

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/dwscript/instances/a1ad-448a-9d56-4154193d49c5/stop
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.6205",
  "error_msg": "The script running history does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.2 资源管理 API

### 5.2.1 创建资源

#### 功能介绍

创建一个新资源，作业节点类型MRS Spark、MRS MapReduce、DLI Spark可以通过资源引用JAR、properties等文件。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
POST /v1/{project\_id}/resources
- 参数说明

表 5-29 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 5-30 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-31 Resource 参数说明

参数名	是否必选	参数类型	说明
name	是	String	资源名称，只能包含英文字母、数字、中文字符、下划线或中划线，且长度为1-32个字符。
type	是	String	资源类型，包含： <ul style="list-style-type: none"> <li>• archive</li> <li>• file</li> <li>• jar</li> </ul>
location	是	String	资源文件所在OBS路径，当类型为jar时，location为主Jar包所在路径。最多1023个字符。示例：obs://myBucket/test.jar
dependFiles	否	List<String>	主Jar包所依赖的JAR包、properties文件。总的长度不能超过10240个字符。
desc	否	String	描述，长度不能超过255个字符。
directory	否	String	资源所在目录。 通过DataArts Studio管理控制台 > 数据开发，左侧列表选择“配置管理 > 资源管理”。在资源的目录树上，可以查看到当前已经创建的目录，默认在根目录/。

## 响应参数

表 5-32 参数说明

参数名	是否必选	参数类型	说明
resourceId	是	String	资源ID

## 请求示例

创建资源名称为test的资源，资源类型为jar，资源文件所在OBS路径为obs://dlf-test/hadoop-mapreduce-examples-2.4.1.jar，资源主Jar包所依赖的JAR包、properties文件为obs://dlf-test/depend1.jar, "obs://dlf-test/depend2.jar，描述为test，资源所在目录为/resource。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources
{
  "name": "test",
  "type": "jar",
  "location": "obs://dlf-test/hadoop-mapreduce-examples-2.4.1.jar",
  "dependFiles": ["obs://dlf-test/depend1.jar","obs://dlf-test/depend2.jar"],
  "desc": "test",
```

```
"directory":"/resource"  
}
```

## 响应示例

- 成功响应

```
{  
  "resourceId":"3624d1c3-5df5-4f20-9af9-98eadad6c5f9"  
}
```

- 失败响应

HTTP状态码 400

```
{  
  "error_code":"DLF.6259",  
  "error_msg":"Files of the same name exist in the directory."  
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.2.2 修改资源

### 功能介绍

此接口可以修改指定资源，修改时，请指定资源ID。

- 资源类型和资源目录不支持修改。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
PUT /v1/{project\_id}/resources/{resource\_id}
- 参数说明

表 5-33 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
resource_id	是	String	资源ID，获取方法请参见 <a href="#">查询资源列表</a> 。返回的id即为resource_id。

## 请求参数

表 5-34 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	<p>工作空间id。</p> <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-35 Resource 参数说明

参数名	是否必选	参数类型	说明
name	是	String	资源名称，只能包含英文字母、数字、中文字符、下划线或中划线，且长度为1-32个字符。
type	是	String	<p>资源类型，包含：</p> <ul style="list-style-type: none"> <li>• archive</li> <li>• file</li> <li>• jar</li> </ul>
location	否	String	资源文件所在OBS路径，当类型为jar时，location为主Jar包所在路径。最多1023个字符。示例：obs://myBucket/test.jar
dependFiles	否	List<String>	主Jar包所依赖的JAR包、properties文件。总的长度不能超过10240个字符。
dependPackages	否	List<DependPackage>	主Jar包所依赖的JAR包、properties文件。总的长度不能超过10240个字符。同时存在dependFiles和dependPackages时，优先解析该字段。
desc	否	String	描述，长度不能超过255个字符。
directory	是	String	<p>资源所在目录。</p> <p>通过DataArts Studio管理控制台 &gt; 数据开发，左侧列表选择“配置管理 &gt; 资源管理”。在资源的目录树上，可以查看到当前已经创建的目录，默认在根目录/。</p>

表 5-36 DependPackage 参数说明

参数名	是否必选	参数类型	说明
type	否	String	文件类型。
location	否	String	文件路径。

## 响应参数

无

## 请求示例

修改资源名称为test的资源内容，资源类型为jar，资源文件所在OBS路径为**obs://dlf-test/hadoop-mapreduce-examples-2.4.1.jar**，资源主Jar包所依赖的JAR包、properties文件为**obs://dlf-test/depend1.jar"**,"obs://dlf-test/depend2.jar"，描述为test，资源所在目录为**/resource**。

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources/3624d1c3-5df5-4f20-9af9-98eadad6c5f9
{
  "name": "test",
  "type": "jar",
  "location": "obs://dlf-test/hadoop-mapreduce-examples-2.4.1.jar",
  "dependFiles": ["obs://dlf-test/depend1.jar","obs://dlf-test/depend2.jar"],
  "desc": "test",
  "directory": "/resource"
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.6241",
  "error_msg": "The resource information does not exist."
}
```

## 5.2.3 查询资源详情

### 功能介绍

查询资源详情，一个资源包含jar、zip、properties等文件，已经创建的资源在DLI Spark、MRS Spark等作业节点中使用。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式



GET /v1/{project\_id}/resources/{resource\_id}

- 参数说明

表 5-37 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
resource_id	是	String	资源ID，获取方法请参见 <a href="#">查询资源列表</a> 。返回的id即为resource_id。

## 请求参数

表 5-38 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

表 5-39 Resource 参数说明

参数名	是否必选	参数类型	说明
name	是	String	资源名称，只能包含英文字母、数字、中文字符、下划线或中划线，且长度为1-32个字符。
type	是	String	资源类型，包含： <ul style="list-style-type: none"> <li>• archive</li> <li>• file</li> <li>• jar</li> </ul>
location	是	String	资源文件所在OBS路径，当类型为jar时，location为主Jar包所在路径。最多256个字符。示例：obs://myBucket/test.jar
dependFiles	否	List<String>	主Jar包所依赖的JAR包、properties文件。总的长度不能超过10240个字符。

参数名	是否必选	参数类型	说明
desc	否	String	描述，长度不能超过255个字符。
directory	否	String	资源所在目录。 通过DataArts Studio管理控制台 > 数据开发，左侧列表选择“配置管理 > 资源管理”。在资源的目录树上，可以查看到当前已经创建的目录，默认在根目录/。

## 请求示例

### 查询资源详情

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources/3624d1c3-5df5-4f20-9af9-98eadad6c5f9
```

## 响应示例

- 成功响应

```
{
  "name": "test",
  "type": "jar",
  "location": "obs://dlf-test/hadoop-mapreduce-examples-2.4.1.jar",
  "dependFiles": ["obs://dlf-test/depend1.jar","obs://dlf-test/depend2.jar"],
  "desc": "test",
  "directory":"/resource"
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code":"DLF.6241",
  "error_msg":"The resource information does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.2.4 删除资源

### 功能介绍

此接口可以删除指定资源。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式

```
DELETE /v1/{project_id}/resources/{resource_id}
```

- 参数说明

表 5-40 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
resource_id	是	String	资源ID，查询方法请参见 <a href="#">查询资源列表</a> 。返回的id即为 resource_id。

## 请求参数

表 5-41 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

无

## 请求示例

```
DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources/3624d1c3-5df5-4f20-9af9-98eadad6c5f9
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.6241",
  "error_msg": "The resource information does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.2.5 查询资源列表

### 功能介绍

查询资源列表。查询时，可指定返回页号和每页的最大记录数。

### 调试

您可以在 [API Explorer](#) 中调试该接口，支持自动认证鉴权。

### URI

- URI 格式  
GET /v1/{project\_id}/resources?  
offset={offset}&limit={limit}&resourceName={resourceName}
- 参数说明

表 5-42 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
offset	否	Integer	分页列表的起始页，默认值为0。取值范围大于等于0。
limit	否	Integer	分页返回结果，指定每页最大记录数，范围[1,100]。 默认值：10
resourceName	否	String	资源名称。

### 请求参数

表 5-43 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

表 5-44 参数说明

参数名	是否必选	参数类型	说明
total	是	Integer	总的资源个数。
resources	是	List<Resource>	资源列表。

表 5-45 Resource 参数说明

参数名	是否必选	参数类型	说明
id	是	String	资源ID，用户查询资源时使用。
name	是	String	资源名称，只能包含英文字母、数字、中文字符、下划线或中划线，且长度为1-32个字符。
type	是	String	资源类型，包含： <ul style="list-style-type: none"> <li>• archive</li> <li>• file</li> <li>• jar</li> </ul>
location	是	String	资源文件所在OBS路径，当类型为jar时，location为主Jar包所在路径。最多256个字符。示例：obs://myBucket/test.jar
dependFiles	否	List<String>	主Jar包所依赖的JAR包、properties文件。总的长度不能超过10240个字符。
desc	否	String	描述，长度不能超过255个字符。
directory	否	String	资源所在目录。 通过DataArts Studio管理控制台 > 数据开发，左侧列表选择“配置管理 > 资源管理”。在资源的目录树上，可以查看到当前已经创建的目录，默认在根目录/。

## 请求示例

查询资源列表

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "total":1,
  "resources":[
    {
      "id":"b384b9e9ab9b4ee8994c8633aabc9505",
      "name":"test",
      "type":"jar",
      "location":"obs://00000000dlf-test/hadoop-mapreduce-examples-2.4.1.jar",
      "dependFiles":[
        "obs://00000000dlf-test/depend1.jar",
        "obs://00000000dlf-test/depend2.jar"
      ],
      "desc":"test",
      "directory":"/resource"
    }
  ]
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code":"DLF.3051",
  "error_msg":"The request parameter is invalid."
}
```

## 5.3 作业开发 API

### 5.3.1 创建作业

#### 功能介绍

此接口可以创建一个新作业，作业由一个或多个相互依赖的节点构成，支持Hive SQL、CDM Job等节点。DLF有两类型作业：批处理作业和实时作业。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
POST /v1/{project\_id}/jobs
- 参数说明

表 5-46 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 5-47 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	<p>工作空间id。</p> <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>多个DataArts Studio实例需要指定工作空间。</li> <li>当工作空间里面没有default工作空间时，工作空间为必填参数，否则会报错。</li> </ul>

表 5-48 参数说明

参数名	是否必选	参数类型	说明
name	是	String	作业名称，只能包含六种字符：英文字母、数字、中文、中划线、下划线和点号，且长度小于等于128个字符。作业名称不能重复。
nodes	是	List<Node>	节点定义，参考 <a href="#">表5-49</a> 。
schedule	是	Schedule 数据结构	调度配置，参考 <a href="#">表5-50</a> 。
params	否	List<Param>	作业参数定义，参考 <a href="#">表5-51</a> 。
directory	否	String	作业在目录树上的路径。创建作业时如果路径目录不存在，会自动创建目录，如/dir/a/，默认在根目录/。
processType	是	String	<p>作业类型:</p> <ul style="list-style-type: none"> <li>REAL_TIME: 实时处理</li> <li>BATCH: 批处理</li> </ul>
singleNodeJobFlag	否	Boolean	是否选择单任务，默认为false。

参数名	是否必选	参数类型	说明
singleNodeJobType	否	String	单任务类型，作业类型为BATCH时类型可选择如下： <ul style="list-style-type: none"> <li>• DLISQL</li> <li>• DWSSQL</li> <li>• HiveSQL</li> <li>• SparkSQL</li> <li>• RDSSQL</li> </ul> 作业类型为REAL_TIME时类型可选择如下：FlinkSQL、FlinkJar、DLISpark。
lastUpdateUser	否	String	作业最后修改人。
logPath	否	String	作业运行日志存放的OBS路径。
basicConfig	否	BasicConfig数据结构	作业的基本信息，参考表5-74。
emptyRunningJob	否	String	取值为0和1。1表示空跑。0表示取消空跑。不设置该参数时，默认为0。
targetStatus	否	String	在开启审批开关后，需要填写该字段。表示创建作业的目标状态，有三种状态：SAVED、SUBMITTED和PRODUCTION，分别表示作业创建后是保存态，提交态，生产态。 <ul style="list-style-type: none"> <li>• 保存态表示作业仅保存，无法调度运行，需要提交并审核通过后才能运行。</li> <li>• 提交态表示作业保存后会自动提交，需要审核通过才能运行。</li> <li>• 生产态表示作业跳过审批环节，创建后可以直接运行。注意：只有工作空间的管理员用户才能创建生产态的作业。审批开关的配置请参见<a href="#">审批中心</a>。</li> </ul>
approvers	否	List<JobApprover>	在开启审批开关后，需要填写该字段，表示作业审批人，参考表5-78。

表 5-49 Node 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	节点名称，只能包含六种字符：英文字母、数字、中文、中划线、下划线和点号，且长度小于等于128个字符。同一个作业中节点名称不能重复。



参数名	是否必选	参数类型	说明
type	是	String	<p>节点类型:</p> <ul style="list-style-type: none"> <li>• HiveSQL: 执行Hive SQL脚本</li> <li>• SparkSQL: 执行Spark SQL脚本</li> <li>• DWSSQL: 执行DWS SQL脚本</li> <li>• DLISQL: 执行DLI SQL脚本</li> <li>• Shell : 执行Shell SQL脚本</li> <li>• CDMJob: 执行CDM作业</li> <li>• DISTransferTask: 创建DIS转储任务</li> <li>• CloudTableManager: CloudTable表管理, 创建和删除表。</li> <li>• OBSManager: OBS路径管理, 包括创建和删除路径。</li> <li>• RestClient: REST API请求</li> <li>• SMN: 发送短信或邮件</li> <li>• MRSSpark: 执行MRS服务的Spark作业</li> <li>• MapReduce: 执行MRS服务的MapReduce作业</li> <li>• MRSFlinkJob: 执行MRS服务的FlinkJob作业。</li> <li>• MRSHetuEngine: 执行MRS服务的HetuEngine作业。</li> <li>• DLISpark: 执行DLF服务的Spark作业</li> <li>• RDSSQL: 传递SQL语句到RDS中执行。</li> <li>• ModelArts Train: 执行ModelArts服务的workflow作业。</li> <li>• Dummy: 空节点作业类型</li> </ul>
location	是	Location数据结构	节点在作业画布上的位置, 参考 <a href="#">表5-52</a> 。
preNodeName	否	List<String>	本节点依赖的前面的节点名称列表。
conditions	否	List<Condition>	节点执行条件, 如果配置此参数, 本节点是否执行由condition的字段expression所保存的EL表达式计算结果决定, 参考 <a href="#">表5-53</a> 。

参数名	是否必选	参数类型	说明
properties	是	List<Property>	<p>节点属性，参考<a href="#">表5-59</a>。</p> <p>每种节点类型有自己的定义，如下：</p> <ul style="list-style-type: none"> <li>• Hive SQL：参考<a href="#">表5-60</a>。</li> <li>• Spark SQL：参考<a href="#">表5-61</a>。</li> <li>• DWS SQL：参考<a href="#">表5-62</a>。</li> <li>• DLI SQL：参考<a href="#">表5-63</a>。</li> <li>• Shell：参考<a href="#">表5-64</a>。</li> <li>• CDM Job：参考<a href="#">表5-65</a>。</li> <li>• DIS TransferTask：参考<a href="#">表5-66</a>。</li> <li>• CloudTableManager：参考<a href="#">表5-67</a>。</li> <li>• OBSManager：参考<a href="#">表5-68</a>。</li> <li>• RestClient：参考<a href="#">表5-69</a>。</li> <li>• SMN：参考<a href="#">表5-70</a>。</li> <li>• MRS Spark：参考<a href="#">表5-71</a>。</li> <li>• MapReduce：参考<a href="#">表5-72</a>。</li> <li>• DLI Spark：参考<a href="#">表5-73</a>。</li> <li>• MRS Flink：参考<a href="#">表5-75</a>。</li> <li>• MRS HetuEngine：参考<a href="#">表5-76</a>。</li> <li>• ModelArts Train：参考<a href="#">表5-77</a>。</li> </ul>
pollingInterval	否	Int	<p>轮询节点执行结果时间间隔。</p> <p>单位：秒，取值范围[1, 60]</p> <p>默认值：10</p>
execTimeOutRetry	否	String	节点是否超时重试，默认值为"false"。
maxExecutionTime	否	Int	<p>节点最大执行时间，如果节点在最大执行时间内还未执行完成，会把节点置为失败状态。</p> <p>单位：分钟，取值范围[5, 7200]，说明：其他值实际无法生效。</p> <p>默认值：60</p>
retryTimes	否	Int	节点失败重试次数。取值范围[1, 100]。 默认值：1
retryInterval	否	Int	<p>失败重试时间间隔。取值范围[5, 600]</p> <p>单位：秒</p> <p>默认值：120</p>

参数名	是否必选	参数类型	说明
failPolicy	否	String	节点失败策略: <ul style="list-style-type: none"> <li>● FAIL: 终止当前作业执行计划</li> <li>● IGNORE: 继续执行下一个节点</li> <li>● SUSPEND: 挂起当前作业执行计划</li> <li>● FAIL_CHILD: 终止后续节点执行计划</li> </ul> 默认值是FAIL。
eventTrigger	否	Event数据结构	实时作业节点事件触发配置, 参考表 5-56。
cronTrigger	否	Cron数据结构	实时作业节点Cron触发配置, 参考表 5-54。

表 5-50 Schedule 数据结构说明

参数名	是否必选	参数类型	说明
type	是	String	调度类型。 <ul style="list-style-type: none"> <li>● EXECUTE_ONCE: 作业立即运行, 只运行一次。</li> <li>● CRON: 作业按指定频率周期执行。</li> <li>● EVENT: 根据事件触发执行。</li> </ul>
cron	否	数据结构	当type为CRON时, 配置调度频率、启动时间等信息, 参考表5-55。
event	否	数据结构	当type为EVENT时, 配置事件源等信息, 参考表5-56。

表 5-51 Param 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	参数名称, 只能包含英文字母、数字、中划线和下划线。不能超过64个字符。
value	是	String	参数值, 不能超过1024个字符。
type	否	String	参数类型。 <ul style="list-style-type: none"> <li>● variable: 变量</li> <li>● constants: 常量</li> </ul> 默认值: variable

表 5-52 Location 数据结构说明

参数名	是否必选	参数类型	说明
x	是	Int	节点在作业画布上的横轴位置。
y	是	Int	节点在作业画布上的纵轴位置。

表 5-53 condition 数据结构说明

参数名	是否必选	参数类型	说明
preNodeName	是	String	本节点依赖的前一个节点名称。
expression	是	String	EL表达式，如果EL表达式的计算结果为true，则触发执行本节点。

表 5-54 CronTrigger 数据结构说明

参数名	是否必选	参数类型	说明
startTime	是	String	调度开始时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。
endTime	否	String	调度结束时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。如果结束时间不配置，作业会按照调度周期一直执行下去。
expression	是	String	Cron表达式，格式为"<秒> <分> <时> <天> <月> <星期>"，每个字段允许的输入值参考表5-57。
expressionTime Zone	否	String	Cron表达式对应的时区信息，例如GMT +8。 默认值：使用DataArts Studio服务端所在的时区。

参数名	是否必选	参数类型	说明
period	是	String	用于描述运行的间隔时间，格式为时间+时间单位。 例如："1 hours","1 days","1 weeks","1 months" 需要与expression中的表达式对应。
dependPrePeriod	否	Boolean	是否依赖本作业上一个运行周期任务的执行结果。 默认值：false
dependJobs	否	DependJobs数据结构	依赖其它作业配置，参考 <a href="#">表5-58</a>
concurrent	否	Integer	支持并发执行数量

表 5-55 Cron 数据结构说明

参数名	是否必选	参数类型	说明
startTime	是	String	调度开始时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。
endTime	否	String	调度结束时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。如果结束时间不配置，作业会按照调度周期一直执行下去。
expression	是	String	Cron表达式，格式为"<秒> <分> <时> <天> <月> <星期>"，每个字段允许的输入值参考 <a href="#">表5-57</a> 。
expressionTime Zone	否	String	Cron表达式对应的时区信息，例如GMT +8。 默认值：使用DataArts Studio服务端所在的时区。
dependPrePeriod	否	Boolean	是否依赖本作业上一个运行周期任务的执行结果。 默认值：false

参数名	是否必选	参数类型	说明
dependJobs	否	DependJobs数据结构	依赖其它作业配置，参考表5-58

表 5-56 Event 数据结构说明

参数名	是否必选	参数类型	说明
eventType	是	String	<p>选择对应的连接名称与topic，当有新的kafka消息时将会触发作业运行一次 配置为：KAFKA</p> <p>事件类型，当前只支持监听DIS通道的新上报数据事件，每上报一条数据，触发作业运行一次。 配置为：DIS</p> <p>选择要监听的OBS路径，如果该路径下有新增文件，则触发调度；新增的文件的路径名，可以通过变量Job.trigger.obsNewFiles引用。前提条件：该OBS路径已经配置DIS消息通知。 配置为：OBS</p>
channel	是	String	<p>DIS通道名称。 通过DIS管理控制台获取通道名称：</p> <ol style="list-style-type: none"> <li>1. 登录管理控制台。</li> <li>2. 单击“数据接入服务”，左侧列表选择“通道管理”。</li> <li>3. 通道管理页面中列出了用户拥有的通道。</li> </ol>
failPolicy	否	String	<p>执行失败处理策略：</p> <ul style="list-style-type: none"> <li>● SUSPEND：挂起</li> <li>● IGNORE：忽略失败，读取下一事件</li> </ul> <p>默认值：SUSPEND</p>
concurrent	否	int	<p>调度并发数 取值范围：1~128 默认值：1</p>

参数名	是否必选	参数类型	说明
readPolicy	否	String	读取策略: <ul style="list-style-type: none"> <li>• LAST : 从上次位置读取</li> <li>• NEW- 从最新位置读取</li> </ul> 默认值 : LAST

表 5-57 Cron 表达式字段取值

字段	取值范围	允许的特殊字符	说明
秒	0-59	, - * /	当前版本只允许输入0。
分	0-59	, - * /	-
时	0-23	, - * /	-
天	1-31	, - * ? / L W C	-
月	1-12	, - * /	当前版本只允许输入*。
星期	1-7	, - * ? / L C #	从星期日开始算起。

表 5-58 DependJobs 数据结构说明

参数名	是否必选	参数类型	说明
jobs	是	List<String >	依赖的作业名称列表，必须依赖已存在的作业。
dependPeriod	否	String	依赖周期: <ul style="list-style-type: none"> <li>• SAME_PERIOD :依赖被依赖作业的同周期任务的执行结果。</li> <li>• PRE_PERIOD :依赖被依赖作业的前一周期的执行结果。</li> </ul> 默认值 : SAME_PERIOD
dependFailPolicy	否	String	依赖作业任务执行失败处理策略: <ul style="list-style-type: none"> <li>• FAIL: 停止作业，设置作业为失败状态</li> <li>• IGNORE: 继续执行作业</li> <li>• SUSPEND : 挂起作业</li> </ul> 默认值 : FAIL

表 5-59 Property 属性说明

参数名	是否必选	参数类型	说明
name	是	String	属性名称
value	是	String	属性值

表 5-60 Hive SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 MRS Hive中的数据库，默认default。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数，key、value形式，多个参数间用\n分隔，例如 key1=value1\nkey2=value2

表 5-61 Spark SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 MRS Spark SQL中的数据库，默认default。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数，key、value形式，多个参数间用\n分隔，例如 key1=value1\nkey2=value2

表 5-62 DWS SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称



参数名	是否必选	参数类型	说明
database	否	String	数据库名称 DWS服务中的数据库，默认postgres。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数，key、value形式，多个参数间用\n分隔，例如key1=value1\nkey2=value2

表 5-63 DLI SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 DLI服务中数据库。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数，key、value形式，多个参数间用\n分隔，例如key1=value1\nkey2=value2

表 5-64 Shell 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
connectionName	是	String	连接名称
arguments	否	String	Shell脚本参数

表 5-65 CDM Job 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	集群名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表可以获取到集群名称。

参数名	是否必选	参数类型	说明
jobName	是	String	作业名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表单击对应的集群管理，进入后在作业管理中获取作业名称。

表 5-66 DISTransferTask 节点属性说明

参数名	是否必选	参数类型	说明
streamName	是	String	DIS通道名称 通过DIS管理控制台获取通道名称： 1. 登录管理控制台。 2. 单击“数据接入服务”，左侧列表选择“通道管理”。 3. 通道管理中列出了用户拥有的通道。
destinationType	是	String	转储目标： <ul style="list-style-type: none"> <li>CloudTable</li> <li>OBS</li> </ul>
duplicatePolicy	是	String	转储任务重名处理策略： <ul style="list-style-type: none"> <li>OVERWRITE：覆盖</li> <li>IGNORE：忽略</li> </ul>
configuration	是	数据结构	转储配置，参考 <a href="#">添加转储任务</a> 的obs_destination_descriptor参数说明和cloudtable_destination_descriptor参数说明。

表 5-67 CloudTableManager 节点属性说明

参数名	是否必选	参数类型	说明
namespace	否	String	命名空间 默认值：default
action	是	String	动作类型： <ul style="list-style-type: none"> <li>CREATE_TABLE：创建表</li> <li>DELETE_TABLE：删除表</li> </ul>
table	否	String	表名

参数名	是否必选	参数类型	说明
columnFamily	否	String	列族

表 5-68 OBSManager 节点属性说明

参数名	是否必选	参数类型	说明
action	是	String	动作类型： <ul style="list-style-type: none"> <li>• CREATE_PATH : 创建OBS路径</li> <li>• DELETE_PATH : 删除OBS路径</li> </ul>
path	是	String	OBS路径

表 5-69 RestClient 节点属性说明

参数名	是否必选	参数类型	说明
url	是	String	URL地址 云服务提供的URL地址。
method	是	String	HTTP方法： <ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> </ul>
headers	否	String	HTTP消息头，每个消息头的格式为<消息头名称>=<值>，多个消息头之间使用换行符分割。
body	否	String	消息体

表 5-70 SMN 节点属性说明

参数名	是否必选	参数类型	说明
topic	是	String	SMN主题URN 通过SMN管理控制台获取SMN主题URN： 1. 登录管理控制台。 2. 单击“消息通知服务”，左侧列表选择“主题管理 > 主题”。 在主题列表中可以获取到SMN主题URN。
subject	是	String	消息标题，给邮箱订阅者发送邮件时作为邮件主题。
messageType	是	String	消息类型 <ul style="list-style-type: none"> <li>• NORMAL</li> <li>• STRUCTURE</li> <li>• TEMPLATE</li> </ul>
message	是	String	发送的消息。

表 5-71 MRS Spark 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
resourcePath	是	String	自定义Spark Jar包OBS资源路径
parameters	是	String	Spark Jar包自定义参数 对于开发的自定义Jar包，可以在此处对参数进行输入替换
input	否	String	输入路径 MRS Spark作业输入数据路径，可以为HDFS或者是OBS路径。

参数名	是否必选	参数类型	说明
output	否	String	输出路径 MRS Spark作业输出数据路径，可以为HDFS或者是OBS路径。
programParameter	否	String	运行程序参数 允许多个key:value，多个参数要用竖线隔开。

表 5-72 MapReduce 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
resourcePath	是	String	资源路径
parameters	是	String	MapReduce作业参数
input	是	String	输入路径 MapReduce作业输入数据路径，可以为HDFS或者是OBS路径。
output	是	String	输出路径 MapReduce作业输出数据路径，可以为HDFS或者是OBS路径。

表 5-73 DLI Spark 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	DLI队列名称 通过DLI管理控制台获取队列名称： 1. 登录管理控制台。 2. 单击“数据湖探索服务”，单击“队列管理”。 在队列管理列表中可以获取到队列名称。
jobName	是	String	DLI作业名称 通过DLI管理控制台获取作业名称： 1. 登录管理控制台。 2. 单击“数据湖探索服务”，单击Spark作业。 3. 选择“作业管理”。 在作业管理列表中可以获取到作业名称。
resourceType	否	String	DLI作业运行资源类型，可选 1. OBS路径：OBS 2. DLI程序包：DLIResources
jobClass	否	String	主类名称。当应用程序类型为“.jar”时，主类名称不能为空。
resourcePath	是	String	JAR包资源路径
jarArgs	否	String	主类入口参数
sparkConfig	否	String	Spark作业运行参数

表 5-74 BasicConfig 作业基本信息说明

参数名	是否必选	参数类型	说明
owner	否	String	作业责任人，长度不能超过128个字符。
agency	否	String	作业委托的名称
isIgnoreWaiting	否	int	实例超时是否忽略等待时间，取值范围为0和1，默认值为1 0：表示实例超时不忽略等待时间 1：表示实例超时忽略等待时间
priority	否	int	作业优先级，取值范围[0, 2]，默认值是0。0代表高优先级，1代表中优先级，2代表低优先级。

参数名	是否必选	参数类型	说明
executeUser	否	String	作业执行用户，必须是已存在的用户名。
instanceTimeout	否	int	实例超时时间，单位是分钟，取值范围 [5, 1440]，默认值是60
customFields	否	Map<String,String>	用户自定义字段，长度不能超过2048个字符。

表 5-75 MRS Flink 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce 服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获得到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
flinkJobType	是	String	客户选择是Flink SQL作业或Flink JAR作业。
flinkJobProcessType	是	String	客户选择批处理模式或流处理模式。
scriptName	否	String	客户选择Flink SQL关联的SQL脚本。
resourcePath	否	String	自定义Flink Jar包OBS资源路径
input	否	String	输入路径 MRS Flink作业输入数据路径，可以为HDFS或者是OBS路径。

参数名	是否必选	参数类型	说明
output	否	String	输出路径 MRS Flink作业输出数据路径，可以为HDFS或者是OBS路径。
programParameter	否	String	运行程序参数 允许多个key:value，多个参数要用竖线隔开。

表 5-76 MRS HetuEngine 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
statementOrScript	是	String	选择使用SQL语句和关联SQL脚本。
scriptName	否	String	选择关联的SQL脚本。
statement	否	String	自定义的SQL内容。
Data Warehouse	是	String	指定HetuEngine服务所需数据连接。
Schema	是	String	使用HetuEngine服务所要访问的数据源schema名称。



参数名	是否必选	参数类型	说明
Database	是	String	使用HetuEngine服务所要访问的数据源database名称。
Queue	否	String	使用HetuEngine服务所需资源队列名称。

表 5-77 ModelArts Train 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称。 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，在左侧列表单击“MRS集群 > 现有集群”。 在现有列表中可以获得到集群名称。
jobName	是	String	MRS作业名称。 用户可自定义。
statementOrScript	是	String	选择使用SQL语句和关联SQL脚本。
scriptName	否	String	选择关联的SQL脚本。

表 5-78 审批人属性说明

参数名	是否必选	参数类型	说明
approverName	是	String	审批人名称。

## 响应参数

无

## 请求示例

创建一个作业名称为myJob的作业，作业类型为BATCH，作业调度配置为CRON，作业在目录树上的路径为/myDir，作业运行日志存放的OBS路径为obs://dlf-test-log。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs
{
  "basicConfig": {
    "customFields": {},
    "executeUser": "",
    "instanceTimeout": 0,
    "owner": "test_user",
    "priority": 0
  },
  "directory": "/myDir",
  "logPath": "obs://dlf-test-log",
  "name": "myJob",
  "nodes": [
    {
      "failPolicy": "FAIL_CHILD",
      "location": {
        "x": "-45.5",
        "y": "-134.5"
      },
      "maxExecutionTime": 360,
      "name": "MRS_Hive_SQL",
      "pollingInterval": 20,
      "preNodeName": [],
      "properties": [
        {
          "name": "scriptName",
          "value": "test_hive_sql"
        },
        {
          "name": "connectionName",
          "value": "mrs_hive_test"
        },
        {
          "name": "database",
          "value": "default"
        },
        {
          "name": "scriptArgs",
          "value": "test_var=111"
        }
      ],
      "retryInterval": 120,
      "retryTimes": 0,
      "type": "HiveSQL"
    }
  ],
  "processType": "BATCH",
  "schedule": {
    "type": "CRON"
  }
}
```

开启审批开关的情况下，创建作业的示例。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs
{
  "basicConfig": {
    "customFields": {},
    "executeUser": "",
    "instanceTimeout": 0,
    "owner": "test_user",
    "priority": 0
  },
  "directory": "/myDir",
```

```
"logPath": "obs://dlf-test-log",
"name": "myJob",
"nodes": [
  {
    "failPolicy": "FAIL_CHILD",
    "location": {
      "x": "-45.5",
      "y": "-134.5"
    },
    "maxExecutionTime": 360,
    "name": "MRS_Hive_SQL",
    "pollingInterval": 20,
    "preNodeName": [],
    "properties": [
      {
        "name": "scriptName",
        "value": "test_hive_sql"
      },
      {
        "name": "connectionName",
        "value": "mrs_hive_test"
      },
      {
        "name": "database",
        "value": "default"
      },
      {
        "name": "scriptArgs",
        "value": "test_var=111"
      }
    ],
    "retryInterval": 120,
    "retryTimes": 0,
    "type": "HiveSQL"
  }
],
"processType": "BATCH",
"schedule": {
  "type": "CRON"
},
"targetStatus": "SUBMITTED",
"approvers": [
  {
    "approverName": "userName1"
  },
  {
    "approverName": "userName2"
  }
]
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.0102",
  "error_msg": "The job name already exists."
}
```

## 5.3.2 修改作业

### 功能介绍

此接口可以对一个作业进行编辑。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
PUT /v1/{project\_id}/jobs/{job\_name}
- 参数说明

表 5-79 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

### 请求参数

表 5-80 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-81 参数说明

参数名	是否必选	参数类型	说明
name	是	String	作业名称，只能包含六种字符：英文字母、数字、中文、中划线、下划线和点号，且长度小于等于128个字符。作业名称不能重复。

参数名	是否必选	参数类型	说明
nodes	是	List<Node>	节点定义，参考 <a href="#">表5-82</a> 。
schedule	是	Schedule 数据结构	调度配置，参考 <a href="#">表5-83</a> 。
params	否	List<Param>	作业参数定义，参考 <a href="#">表5-84</a> 。
directory	否	String	作业在目录树上的路径。创建作业时如果路径目录不存在，会自动创建目录，如/dir/a/，默认在根目录/。
cleanOverdueDays	否	Int	设置作业的最大超时时间。
cleanWaitingJob	否	String	清除等待的作业。
emptyRunningJob	否	String	取值为0和1,1表示空跑,0表示:取消空跑,不设置该参数时,默认为0。
processType	是	String	作业类型: <ul style="list-style-type: none"> <li>REAL_TIME: 实时处理</li> <li>BATCH: 批处理</li> </ul>
id	否	Long	作业Id。获取方法请参见 <a href="#">查询作业详情</a> 。返回的id即为作业id。
lastUpdateUser	否	String	作业最后修改人。
logPath	否	String	作业运行日志存放的OBS路径。
basicConfig	否	BasicConfig 数据结构	作业的基本信息，参考 <a href="#">表5-107</a> 。
targetStatus	否	String	在开启审批开关后，需要填写该字段。表示创建作业的目标状态，有三种状态：SAVED、SUBMITTED和PRODUCTION，分别表示作业创建后是保存态，提交态，生产态。 <ul style="list-style-type: none"> <li>保存态表示作业仅保存，无法调度运行，需要提交并审核通过后才能运行。</li> <li>提交态表示作业保存后会自动提交，需要审核通过才能运行。</li> <li>生产态表示作业跳过审批环节，创建后可以直接运行。注意：只有工作空间的管理员用户才能创建生产态的作业。审批开关的配置请参见<a href="#">审批中心</a>。</li> </ul>

参数名	是否必选	参数类型	说明
approvers	否	List<JobApprover>	在开启审批开关后，需要填写该字段，表示作业审批人，参考表5-111。 审批开关的配置请参见 <a href="#">审批中心</a> 。

表 5-82 Node 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	节点名称，只能包含六种字符：英文字母、数字、中文、中划线、下划线和点号，且长度小于等于128个字符。同一个作业中节点名称不能重复。
type	是	String	节点类型： <ul style="list-style-type: none"> <li>• HiveSQL：执行Hive SQL脚本</li> <li>• SparkSQL：执行Spark SQL脚本</li> <li>• DWSSQL：执行DWS SQL脚本</li> <li>• DLISQL：执行DLI SQL脚本</li> <li>• Shell：执行Shell SQL脚本</li> <li>• CDMJob：执行CDM作业</li> <li>• DISTransferTask：创建DIS转储任务</li> <li>• CloudTableManager：CloudTable表管理，创建和删除表。</li> <li>• OBSManager：OBS路径管理，包括创建和删除路径。</li> <li>• RestClient：REST API请求</li> <li>• SMN：发送短信或邮件</li> <li>• MRSSpark：执行MRS服务的Spark作业</li> <li>• MapReduce：执行MRS服务的MapReduce作业</li> <li>• MRSFlinkJob：执行MRS服务的FlinkJob作业。</li> <li>• MRSHetuEngine：执行MRS服务的HetuEngine作业。</li> <li>• DLISpark：执行DLF服务的Spark作业</li> <li>• RDSSQL：传递SQL语句到RDS中执行。</li> <li>• ModelArts Train：执行ModelArts服务的workflow作业。</li> </ul>

参数名	是否必选	参数类型	说明
location	是	Location数据结构	节点在作业画布上的位置，参考表5-85。
preNodeName	否	List<String>	本节点依赖的前面的节点名称列表。
conditions	否	List<Condition>	节点执行条件，如果配置此参数，本节点是否执行由condition的字段expression所保存的EL表达式计算结果决定,参考表5-86。
properties	是	List<Property>	节点属性，参考表5-92。 每种节点类型有自己的定义，如下： <ul style="list-style-type: none"> <li>• Hive SQL：参考表5-93。</li> <li>• Spark SQL：参考表5-94。</li> <li>• DWS SQL：参考表5-95。</li> <li>• DLI SQL：参考表5-96。</li> <li>• Shell：参考表5-97。</li> <li>• CDM Job：参考表5-98。</li> <li>• DIS TransferTask：参考表5-99。</li> <li>• CloudTableManager：参考表5-100。</li> <li>• OBSManager - 参考表5-101。</li> <li>• RESTAPI：参考表5-102。</li> <li>• SMN：参考表5-103。</li> <li>• MRS Spark：参考表5-104。</li> <li>• MapReduce：参考表5-105。</li> <li>• DLI Spark：参考表5-106。</li> <li>• MRS Flink：参考表5-108。</li> <li>• MRS HetuEngine：参考表5-109。</li> <li>• ModelArts Train：参考表5-110。</li> </ul>
pollingInterval	否	Int	轮询节点执行结果时间间隔。 单位：秒，取值范围[1, 60] 默认值: 10
execTimeOutRetry	否	String	节点是否超时重试，默认值为"false"。
maxExecutionTime	否	Int	节点最大执行时间，如果节点在最大执行时间内还未执行完成，会把节点置为失败状态。 单位：分钟，取值范围[5, 1440] 默认值：60

参数名	是否必选	参数类型	说明
retryTimes	否	Int	节点失败重试次数。取值范围[1, 100]。 默认值：1
retryInterval	否	Int	失败重试时间间隔。取值范围[5, 600] 单位：秒 默认值：120
failPolicy	否	String	节点失败策略： <ul style="list-style-type: none"> <li>FAIL：终止当前作业执行计划</li> <li>IGNORE：继续执行下一个节点</li> <li>SUSPEND：挂起当前作业执行计划</li> <li>FAIL_CHILD：终止后续节点执行计划</li> </ul> 默认值是FAIL。
eventTrigger	否	Event数据结构	实时作业节点事件触发配置，参考表5-89。
cronTrigger	否	Cron数据结构	实时作业节点Cron触发配置，参考表5-87。

表 5-83 Schedule 数据结构说明

参数名	是否必选	参数类型	说明
type	是	String	调度类型。 <ul style="list-style-type: none"> <li>EXECUTE_ONCE：作业立即运行，只运行一次。</li> <li>CRON：作业按指定频率周期执行。</li> <li>EVENT：根据事件触发执行。</li> </ul>
cron	否	数据结构	当type为CRON时，配置调度频率、启动时间等信息，参考表5-88。
event	否	数据结构	当type为EVENT时，配置事件源等信息，参考表5-89。

表 5-84 Param 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	参数名称，只能包含英文字母、数字、中划线和下划线。不能超过64个字符。



参数名	是否必选	参数类型	说明
value	是	String	参数值，不能超过1024个字符。
type	否	String	参数类型 <ul style="list-style-type: none"> <li>• variable: 变量</li> <li>• constants: 常量</li> </ul> 默认值：variable

表 5-85 Location 数据结构说明

参数名	是否必选	参数类型	说明
x	是	Int	节点在作业画布上的横轴位置。
y	是	Int	节点在作业画布上的纵轴位置。

表 5-86 condition 数据结构说明

参数名	是否必选	参数类型	说明
preNodeName	是	String	本节点依赖的前一个节点名称。
expression	是	String	EL表达式，如果EL表达式的计算结果为true，则触发执行本节点。

表 5-87 CronTrigger 数据结构说明

参数名	是否必选	参数类型	说明
startTime	是	String	调度开始时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。
endTime	否	String	调度结束时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。如果结束时间不配置，作业会按照调度周期一直执行下去。

参数名	是否必选	参数类型	说明
expression	是	String	Cron表达式, 格式为"<秒> <分> <时> <天> <月> <星期>", 每个字段允许的输入值参考表5-90。
expressionTime Zone	否	String	Cron表达式对应的时区信息, 例如GMT +8。 默认值: 使用DataArts Studio服务端所在的时区。
period	是	String	用于描述运行的间隔时间, 格式为时间+时间单位。 例如: "1 hours","1 days","1 weeks","1 months" 需要与expression中的表达式对应。
dependPrePeriod	否	Boolean	是否依赖本作业上一个运行周期任务的执行结果。 默认值: false
dependJobs	否	DependJobs数据结构	依赖其它作业配置, 参考表5-91。
concurrent	否	Integer	支持并发执行数量。

表 5-88 Cron 数据结构说明

参数名	是否必选	参数类型	说明
startTime	是	String	调度开始时间, 采用ISO 8601时间表示方法, 格式为yyyy-MM-dd'T'HH:mm:ssZ, 例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒, 在正8区, 即北京时区。
endTime	否	String	调度结束时间, 采用ISO 8601时间表示方法, 格式为yyyy-MM-dd'T'HH:mm:ssZ, 例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒, 在正8区, 即北京时区。如果结束时间不配置, 作业会按照调度周期一直执行下去。
expression	是	String	Cron表达式, 格式为"<秒> <分> <时> <天> <月> <星期>", 每个字段允许的输入值参考表5-90。

参数名	是否必选	参数类型	说明
expressionTime Zone	否	String	Cron表达式对应的时区信息，例如GMT +8。 默认值：使用DataArts Studio服务端所在的时区。
dependPrePeriod	否	Boolean	是否依赖本作业上一个运行周期任务的执行结果。 默认值：false
dependJobs	否	DependJobs数据结构	依赖其它作业配置，参考 <a href="#">表5-91</a>

表 5-89 Event 数据结构说明

参数名	是否必选	参数类型	说明
eventType	是	String	选择对应的连接名称与topic，当有新的kafka消息时将会触发作业运行一次。 配置为：KAFKA 事件类型，当前只支持监听DIS通道的新上报数据事件，每上报一条数据，触发作业运行一次。 配置为：DIS 选择要监听的OBS路径，如果该路径下有新增文件，则触发调度；新增的文件的路径名，可以通过变量Job.trigger.obsNewFiles引用。前提条件：该OBS路径已经配置DIS消息通知。 配置为：OBS
channel	是	String	DIS通道名称。 通过DIS管理控制台获取通道名称： 1. 登录管理控制台。 2. 单击“数据接入服务”，左侧列表选择“通道管理”。 3. 通道管理页面中列出了用户拥有的通道。
failPolicy	否	String	执行失败处理策略： <ul style="list-style-type: none"> <li>● SUSPEND：挂起</li> <li>● IGNORE：忽略失败，读取下一事件。</li> </ul> 默认值：SUSPEND

参数名	是否必选	参数类型	说明
concurrent	否	int	调度并发数。 取值范围：1~128 默认值：1
readPolicy	否	String	读取策略： <ul style="list-style-type: none"> <li>• LAST：从上次位置读取</li> <li>• NEW- 从最新位置读取</li> </ul> 默认：LAST

表 5-90 Cron 表达式字段取值

字段	取值范围	允许的特殊字符	说明
秒	0-59	, - * /	当前版本只允许输入0。
分	0-59	, - * /	-
时	0-23	, - * /	-
天	1-31	, - * ? / L W C	-
月	1-12	, - * /	当前版本只允许输入*。
星期	1-7	, - * ? / L C #	从星期日开始算起。

表 5-91 DependJobs 数据结构说明

参数名	是否必选	参数类型	说明
jobs	是	List<String>	依赖的作业名称列表，必须依赖已存在的作业。
dependPeriod	否	String	依赖周期： <ul style="list-style-type: none"> <li>• SAME_PERIOD :依赖被依赖作业的同周期任务的执行结果。</li> <li>• PRE_PERIOD :依赖被依赖作业的前一周期的任务的执行结果。</li> </ul> 默认值：SAME_PERIOD

参数名	是否必选	参数类型	说明
dependFailPolicy	否	String	依赖作业任务执行失败处理策略: <ul style="list-style-type: none"> <li>• FAIL: 停止作业, 设置作业为失败状态。</li> <li>• IGNORE: 继续执行作业</li> <li>• SUSPEND: 挂起作业</li> </ul> 默认值: FAIL

表 5-92 Property 属性说明

参数名	是否必选	参数类型	说明
name	是	String	属性名称。
value	是	String	属性值。

表 5-93 Hive SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称。
database	否	String	数据库名称。 MRS Hive中的数据库, 默认default。
connectionName	否	String	连接名称。
scriptArgs	否	String	脚本参数, key、value形式, 多个参数间用\n分隔, 例如: key1=value1\nkey2=value2。

表 5-94 Spark SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称。
database	否	String	数据库名称。 MRS Spark SQL中的数据库, 默认default。

参数名	是否必选	参数类型	说明
connectionName	否	String	连接名称。
scriptArgs	否	String	脚本参数, key、value形式, 多个参数间用\n分隔, 例如: key1=value1\nkey2=value2。

表 5-95 DWS SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 DWS服务中的数据库, 默认postgres。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数, key、value形式, 多个参数间用\n分隔, 例如key1=value1\nkey2=value2

表 5-96 DLI SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 DLI服务中数据库。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数, key、value形式, 多个参数间用\n分隔, 例如key1=value1\nkey2=value2

表 5-97 Shell 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称

参数名	是否必选	参数类型	说明
connectionName	是	String	连接名称
arguments	否	String	Shell脚本参数

表 5-98 CDM Job 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	集群名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表可以获取到集群名称。
jobName	是	String	作业名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表单击对应的集群管理，进入后在作业管理中获取作业名称。

表 5-99 DISTransferTask 节点属性说明

参数名	是否必选	参数类型	说明
streamName	是	String	DIS通道名称。 通过DIS管理控制台获取通道名称： 1. 登录管理控制台。 2. 单击“数据接入服务”，左侧列表选择“通道管理”。 3. 通道管理中列出了用户拥有的通道。
destinationType	是	String	转储目标： <ul style="list-style-type: none"> <li>CloudTable</li> <li>OBS</li> </ul>
duplicatePolicy	是	String	转储任务重名处理策略： <ul style="list-style-type: none"> <li>OVERWRITE: 覆盖</li> <li>IGNORE: 忽略</li> </ul>

参数名	是否必选	参数类型	说明
configuration	是	数据结构	转储配置，参考 <a href="#">添加转储任务</a> 的 obs_destination_descriptor 参数说明和 cloudtable_destination_descriptor 参数说明。

表 5-100 CloudTableManager 节点属性说明

参数名	是否必选	参数类型	说明
namespace	否	String	命名空间。 默认值：default
action	是	String	动作类型： <ul style="list-style-type: none"> <li>CREATE_TABLE：创建表</li> <li>DELETE_TABLE：删除表</li> </ul>
table	否	String	表名。
columnFamily	否	String	列族。

表 5-101 OBSManager 节点属性说明

参数名	是否必选	参数类型	说明
action	是	String	动作类型： <ul style="list-style-type: none"> <li>CREATE_PATH：创建OBS路径</li> <li>DELETE_PATH：删除OBS路径</li> </ul>
path	是	String	OBS路径

表 5-102 RESTAPI 节点属性说明

参数名	是否必选	参数类型	说明
url	是	String	URL地址。 云服务提供的URL地址。



参数名	是否必选	参数类型	说明
method	是	String	HTTP方法： <ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> </ul>
headers	否	String	HTTP消息头，每个消息头的格式为<消息头名称>=<值>，多个消息头之间使用换行符分割。
body	否	String	消息体。

表 5-103 SMN 节点属性说明

参数名	是否必选	参数类型	说明
topic	是	String	SMN主题URN。 通过SMN管理控制台获取SMN主题URN： <ol style="list-style-type: none"> <li>1. 登录管理控制台。</li> <li>2. 单击“消息通知服务”，左侧列表选择“主题管理 &gt; 主题”。</li> </ol> 在主题列表中可以获取到SMN主题URN。
subject	是	String	消息标题，给邮箱订阅者发送邮件时作为邮件主题。
messageType	是	String	消息类型： <ul style="list-style-type: none"> <li>• NORMAL</li> <li>• STRUCTURE</li> <li>• TEMPLATE</li> </ul>
message	是	String	发送的消息。

表 5-104 MRS Spark 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称。 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称。 用户自定义。
resourcePath	是	String	自定义Spark Jar包OBS资源路径。
parameters	是	String	Spark Jar包自定义参数。 对于开发的自定义Jar包，可以在此处对参数进行输入替换。
input	否	String	输入路径。 MRS Spark作业输入数据路径，可以为HDFS或者是OBS路径。
output	否	String	输出路径。 MRS Spark作业输出数据路径，可以为HDFS或者是OBS路径。
programParameter	否	String	运行程序参数。 允许多个key:value，多个参数要用竖线隔开。

表 5-105 MapReduce 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
resourcePath	是	String	资源路径

参数名	是否必选	参数类型	说明
parameters	是	String	MapReduce作业参数
input	是	String	输入路径 MapReduce作业输入数据路径，可以为HDFS或者是OBS路径。
output	是	String	输出路径 MapReduce作业输出数据路径，可以为HDFS或者是OBS路径。

表 5-106 DLI Spark 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	DLI队列名称 通过DLI管理控制台获取队列名称： 1. 登录管理控制台。 2. 单击“数据湖探索服务”，单击“队列管理”。 在队列管理列表中可以获取到队列名称。
jobName	是	String	DLI作业名称 通过DLI管理控制台获取作业名称： 1. 登录管理控制台。 2. 单击“数据湖探索服务”，单击Spark作业。 3. 选择“作业管理”。 在作业管理列表中可以获取到作业名称。
resourceType	否	String	DLI作业运行资源类型，可选 1. OBS路径：OBS 2. DLI程序包：DLIResources
jobClass	否	String	主类名称。当应用程序类型为“.jar”时，主类名称不能为空。
resourcePath	是	String	JAR包资源路径
jarArgs	否	String	主类入口参数
sparkConfig	否	String	Spark作业运行参数

表 5-107 BasicConfig 作业基本信息说明

参数名	是否必选	参数类型	说明
owner	否	String	作业责任人，长度不能超过128个字符。
priority	否	int	作业优先级，取值范围[0, 2]，默认值是0。0代表高优先级，1代表中优先级，2代表低优先级。
executeUser	否	String	作业执行用户，必须是已存在的用户名。
instanceTimeout	否	int	实例超时时间，单位是分钟，取值范围 [5, 1440]，默认值是60
customFields	否	Map<String,String>	用户自定义字段，长度不能超过2048个字符。

表 5-108 MRS Flink 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce 服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
flinkJobType	是	String	客户选择是Flink SQL作业或Flink JAR作业。
flinkJobProcessType	是	String	客户选择批处理模式或流处理模式。
scriptName	否	String	客户选择Flink SQL 关联的SQL脚本。
resourcePath	否	String	自定义Flink Jar包 OBS资源路径

参数名	是否必选	参数类型	说明
input	否	String	输入路径 MRS Flink作业输入数据路径，可以为HDFS或者是OBS路径。
output	否	String	输出路径 MRS Flink作业输出数据路径，可以为HDFS或者是OBS路径。
programParameter	否	String	运行程序参数 允许多个key:value，多个参数要用竖线隔开。

表 5-109 MRS HetuEngine 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获得到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
statementOrScript	是	String	选择使用SQL语句和关联SQL脚本。
scriptName	否	String	选择关联的SQL脚本。
statement	否	String	自定义的SQL内容。

参数名	是否必选	参数类型	说明
Data Warehouse	是	String	指定HetuEngine服务所需数据连接。
Schema	是	String	使用HetuEngine服务所要访问的数据源schema名称。
Database	是	String	使用HetuEngine服务所要访问的数据源database名称。
Queue	否	String	使用HetuEngine服务所需资源队列名称。

表 5-110 ModelArts Train 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称。 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，在左侧列表单击“MRS集群 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称。 用户可自定义。
statementOrScript	是	String	选择使用SQL语句和关联SQL脚本。
scriptName	否	String	选择关联的SQL脚本。

表 5-111 审批人属性说明

参数名	是否必选	参数类型	说明
approverName	是	String	审批人名称。

## 请求示例

修改作业名称为myJob的作业属性，作业id为608627，作业最后修改人为test\_user，作业类型为BATCH，作业调度配置为CRON，作业在目录树上的路径为/myDir，作业运行日志存放的OBS路径为obs://dlf-test-log。

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob
```

```
{
  "basicConfig": {
    "customFields": {},
    "executeUser": "",
    "instanceTimeout": 0,
    "owner": "test_user",
    "priority": 0
  },
  "directory": "/myDir",
  "id": 608627,
  "lastUpdateUser": "test_user",
  "logPath": "obs://dlf-test-log",
  "name": "myJob",
  "nodes": [
    {
      "failPolicy": "FAIL_CHILD",
      "location": {
        "x": "-45.5",
        "y": "-134.5"
      },
      "maxExecutionTime": 360,
      "name": "MRS_Hive_SQL",
      "pollingInterval": 20,
      "preNodeName": [],
      "properties": [
        {
          "name": "scriptName",
          "value": "test_hive_sql"
        },
        {
          "name": "connectionName",
          "value": "mrs_hive_test"
        },
        {
          "name": "database",
          "value": "default"
        },
        {
          "name": "scriptArgs",
          "value": "test_var=111"
        }
      ],
      "retryInterval": 120,
      "retryTimes": 0,
      "type": "HiveSQL"
    }
  ],
  "processType": "BATCH",
  "schedule": {
    "type": "CRON"
  }
}
```

开启审批开关的情况下，修改作业myJob的属性。

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob
```

```
{
  "basicConfig": {
    "customFields": {},
    "executeUser": "",
    "instanceTimeout": 0,
    "owner": "test_user",
    "priority": 0
  }
```

```
},
"directory": "/myDir",
"id": 608627,
"lastUpdateUser": "test_user",
"logPath": "obs://dlf-test-log",
"name": "myJob",
"nodes": [
  {
    "failPolicy": "FAIL_CHILD",
    "location": {
      "x": "-45.5",
      "y": "-134.5"
    },
    "maxExecutionTime": 360,
    "name": "MRS_Hive_SQL",
    "pollingInterval": 20,
    "preNodeName": [],
    "properties": [
      {
        "name": "scriptName",
        "value": "test_hive_sql"
      },
      {
        "name": "connectionName",
        "value": "mrs_hive_test"
      },
      {
        "name": "database",
        "value": "default"
      },
      {
        "name": "scriptArgs",
        "value": "test_var=111"
      }
    ],
    "retryInterval": 120,
    "retryTimes": 0,
    "type": "HiveSQL"
  }
],
"processType": "BATCH",
"schedule": {
  "type": "CRON"
},
"approvers": [
  {
    "approverName": "userName1"
  },
  {
    "approverName": "userName2"
  }
]
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF0100",
  "error_msg": "The job does not exists."
}
```



### 5.3.3 查询作业列表

#### 功能介绍

查询批处理或实时作业列表，每次查询返回的作业数量上限是100。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
GET /v1/{project\_id}/jobs?  
jobType={jobType}&offset={offset}&limit={limit}&jobName={jobName}&tags={tags}
- 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
jobType	否	String	作业类型： <ul style="list-style-type: none"> <li>• REAL_TIME：实时处理</li> <li>• BATCH：批处理</li> </ul> 默认值：BATCH
offset	否	Integer	分页列表的页数，默认值为0。取值范围大于等于0。
limit	否	Integer	分页返回结果，指定每页最大记录数。范围[1,1000] 默认值：10
jobName	否	String	作业名称
tags	否	String	作业标签列表，如果有多个标签，以英文逗号分隔。例如：tag1,tag2,tag3。

## 请求参数

表 5-112 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

## 响应参数

表 5-113 参数说明

参数名	是否必选	参数类型	说明
total	是	Integer	作业数量。
jobs	是	List<Job>	作业列表，参考 <a href="#">表5-114</a> 。

表 5-114 Job 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	作业名称。
jobType	是	String	作业类型。 • REAL_TIME：实时处理 • BATCH：批处理
owner	否	String	作业责任人，长度不能超过128个字符。
priority	否	int	作业优先级，取值范围[0, 2]，默认值是0。0代表高优先级，1代表中优先级，2代表低优先级。

参数名	是否必选	参数类型	说明
status	是	String	作业状态。 当jobType为REAL_TIME时, 有如下状态: <ul style="list-style-type: none"> <li>STARTING: 启动中</li> <li>NORMAL: 运行中</li> <li>EXCEPTION: 启动或停止异常</li> <li>STOPPING: 停止中</li> <li>STOPPED: 已停止</li> </ul> 当jobType为BATCH时, 有如下状态: <ul style="list-style-type: none"> <li>SCHEDULING: 调度中</li> <li>STOPPED: 已停止调度</li> <li>PAUSED: 暂停调度</li> </ul>
createUser	是	String	作业的创建者。
createTime	是	Long	作业的创建时间。
startTime	否	Long	作业的启动时间。
endTime	否	Long	作业配置的结束时间。
lastInstanceStatus	否	String	作业最近一次运行实例状态, 当jobType为BATCH时才有本字段。
lastInstanceEndTime	否	Long	作业最近一次运行实例运行结束时间, 当jobType为BATCH时才有本字段。
lastUpdateTime	否	Long	作业最后一次更新时间。
lastUpdateUser	否	String	作业最后一次更新用户。
path	否	String	作业的路径。
singleNodeJobFlag	否	boolean	作业是否为单任务作业。

## 请求示例

查询作业列表。

```
GET /v1/ff6b627b9d7b45b48f773be511c1a2b8/jobs
```

## 响应示例

- 成功响应

```
{
  "jobs": [
    {
      "createTime": 1680783636000,
      "createUser": "test_user",
    }
  ]
}
```

```

        "jobType": "BATCH",
        "lastInstanceEndTime": 1682519407000,
        "lastInstanceStatus": "success",
        "lastUpdateUser": "test_user",
        "name": "myJob",
        "owner": "test_user",
        "path": "/",
        "priority": 0,
        "singleNodeJobFlag": false,
        "status": "SCHEDULING"
    }
  ],
  "total": 1
}

```

- 失败响应

HTTP状态码 400

```

{
  "error_code": "DLF3051",
  "error_msg": "The request parameter is invalid."
}

```

### 5.3.4 查询作业详情

#### 功能介绍

此接口可以查询指定DLF作业定义。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
GET /v1/{project\_id}/jobs/{job\_name}?version={version}
- 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。
version	否	Integer	作业版本号，若传入版本号，则查询指定版本号的作业；若不传入，则查询最新的版本作业。

## 请求参数

表 5-115 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	<p>工作空间id。</p> <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

表 5-116 参数说明

参数名	是否必选	参数类型	说明
name	是	String	作业名称，只能包含六种字符：英文字母、数字、中文、中划线、下划线和点号，且长度小于等于128个字符。作业名称不能重复。
nodes	是	List<Node>	节点定义，参考 <a href="#">表5-117</a>
schedule	是	Schedule 数据结构	调度配置，参考 <a href="#">表5-118</a>
params	否	List<Param>	作业参数定义，参考 <a href="#">表5-119</a>
directory	否	String	作业在目录树上的路径。创建作业时如果路径目录不存在，会自动创建目录，如/dir/a/，默认在根目录/。
processType	是	String	<p>作业类型:</p> <ul style="list-style-type: none"> <li>REAL_TIME：实时处理</li> <li>BATCH：批处理</li> </ul>
id	否	Long	作业Id。用户查询作业时使用。
createTime	否	Long	作业创建时间。
downstreamJobs	否	String	下游依赖当前作业的作业名称。
lastUpdateUser	否	String	作业最后修改人。
logPath	否	String	作业运行日志存放的OBS路径。

参数名	是否必选	参数类型	说明
basicConfig	否	BasicConfig数据结构	作业的基本信息，参考 <a href="#">表5-142</a>
description	否	String	作业描述信息
cleanoverdueDays	否	integer	设置作业的最大超时时间
cleanWaitingJob	否	String	清除等待的作业
emptyRunningJob	否	String	是否空跑
version	否	String	作业版本信息

表 5-117 Node 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	节点名称，只能包含六种字符：英文字母、数字、中文、中划线、下划线和点号，且长度小于等于128个字符。同一个作业中节点名称不能重复。

参数名	是否必选	参数类型	说明
type	是	String	<p>节点类型:</p> <ul style="list-style-type: none"> <li>• HiveSQL: 执行Hive SQL脚本</li> <li>• SparkSQL: 执行Spark SQL脚本</li> <li>• DWSSQL: 执行DWS SQL脚本</li> <li>• DLISQL: 执行DLI SQL脚本</li> <li>• Shell : 执行Shell SQL脚本</li> <li>• CDMJob: 执行CDM作业</li> <li>• DISTransferTask: 创建DIS转储任务</li> <li>• CloudTableManager: CloudTable表管理, 创建和删除表。</li> <li>• OBSManager: OBS路径管理, 包括创建和删除路径。</li> <li>• RestClient: REST API请求</li> <li>• SMN: 发送短信或邮件</li> <li>• MRSSpark: 执行MRS服务的Spark作业</li> <li>• MapReduce: 执行MRS服务的MapReduce作业</li> <li>• MRSFlinkJob: 执行MRS服务的FlinkJob作业。</li> <li>• MRSHetuEngine: 执行MRS服务的HetuEngine作业。</li> <li>• DLISpark: 执行DLF服务的Spark作业</li> <li>• RDSSQL: 传递SQL语句到RDS中执行。</li> <li>• Dummy: 空节点作业类型</li> </ul>
location	是	Location数据结构	节点在作业画布上的位置, 参考 <a href="#">表5-120</a> 。
preNodeName	否	List<String>	本节点依赖的前面的节点名称列表。
conditions	否	List<Condition>	节点执行条件, 如果配置此参数, 本节点是否执行由condition的字段expression所保存的EL表达式计算结果决定,参考 <a href="#">表5-121</a> 。

参数名	是否必选	参数类型	说明
properties	是	List<Property>	节点属性，参考 <a href="#">表5-127</a> 。 每种节点类型有自己的定义，如下： <ul style="list-style-type: none"> <li>• Hive SQL：参考<a href="#">表5-128</a>。</li> <li>• Spark SQL：参考<a href="#">表5-129</a>。</li> <li>• DWS SQL：参考<a href="#">表5-130</a>。</li> <li>• DLI SQL：参考<a href="#">表5-131</a>。</li> <li>• Shell：参考<a href="#">表5-132</a>。</li> <li>• CDM Job：参考<a href="#">表5-133</a>。</li> <li>• DIS TransferTask：参考<a href="#">表5-134</a>。</li> <li>• CloudTableManager：参考<a href="#">表5-135</a>。</li> <li>• OBSManager - 参考<a href="#">表5-136</a>。</li> <li>• RESTAPI：参考<a href="#">表5-137</a>。</li> <li>• SMN：参考<a href="#">表5-138</a>。</li> <li>• MRS Spark：参考<a href="#">表5-139</a>。</li> <li>• MapReduce：参考<a href="#">表5-140</a>。</li> <li>• DLI Spark：参考<a href="#">表5-141</a>。</li> <li>• MRS Flink：参考<a href="#">表5-143</a>。</li> <li>• MRS HetuEngine：参考<a href="#">表5-144</a>。</li> </ul>
pollingInterval	否	Int	轮询节点执行结果时间间隔。 单位：秒，取值范围[1, 60] 默认值：10
execTimeOutRetry	否	String	节点是否超时重试，默认值为"false"
maxExecutionTime	否	Int	节点最大执行时间，如果节点在最大执行时间内还未执行完成，会把节点置为失败状态。 单位：分钟，取值范围[5, 1440] 默认值：60
retryTimes	否	Int	节点失败重试次数。取值范围[0, 5]。0代表不重试 默认值：0
retryInterval	否	Int	失败重试时间间隔。取值范围[5, 120] 单位：秒 默认值：120



参数名	是否必选	参数类型	说明
failPolicy	否	String	节点失败策略: <ul style="list-style-type: none"> <li>● FAIL: 终止当前作业执行计划</li> <li>● IGNORE: 继续执行下一个节点</li> <li>● SUSPEND: 挂起当前作业执行计划</li> <li>● FAIL_CHILD: 终止后续节点执行计划</li> </ul> 默认值是FAIL。
eventTrigger	否	Event数据结构	实时作业节点事件触发配置, 参考表5-124。
cronTrigger	否	Cron数据结构	实时作业节点Cron触发配置, 参考表5-122。

表 5-118 Schedule 数据结构说明

参数名	是否必选	参数类型	说明
type	是	String	调度类型。 <ul style="list-style-type: none"> <li>● EXECUTE_ONCE: 作业立即运行, 只运行一次。</li> <li>● CRON: 作业按指定频率周期执行。</li> <li>● EVENT: 根据事件触发执行。</li> </ul>
cron	否	数据结构	当type为CRON时, 配置调度频率、启动时间等信息, 参考表5-123。
event	否	数据结构	当type为EVENT时, 配置事件源等信息, 参考表5-124。

表 5-119 Param 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	参数名称, 只能包含英文字母、数字、中划线和下划线。不能超过64个字符。
value	是	String	参数值, 不能超过1024个字符。
type	否	String	参数类型 <ul style="list-style-type: none"> <li>● variable: 变量</li> <li>● constants: 常量</li> </ul> 默认值: variable

表 5-120 Location 数据结构说明

参数名	是否必选	参数类型	说明
x	是	Int	节点在作业画布上的横轴位置
y	是	Int	节点在作业画布上的纵轴位置

表 5-121 condition 数据结构说明

参数名	是否必选	参数类型	说明
preNodeName	是	String	本节点依赖的前一个节点名称
expression	是	String	EL表达式，如果EL表达式的计算结果为true，则触发执行本节点。

表 5-122 CronTrigger 数据结构说明

参数名	是否必选	参数类型	说明
startTime	是	String	调度开始时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。
endTime	否	String	调度结束时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。如果结束时间不配置，作业会按照调度周期一直执行下去。
expression	是	String	Cron表达式，格式为"<秒> <分> <时> <天> <月> <星期>"，每个字段允许的输入值参考表5-125。
expressionTime Zone	否	String	Cron表达式对应的时区信息，例如GMT +8。 默认值：使用DataArts Studio服务端所在的时区。

参数名	是否必选	参数类型	说明
period	是	String	用于描述运行的间隔时间，格式为时间+时间单位。 例如："1 hours","1 days","1 weeks","1 months" 需要与expression中的表达式对应。
dependPrePeriod	否	Boolean	是否依赖本作业上一个运行周期任务的执行结果。 默认值：false
dependJob	否	DependJob数据结构	依赖其它作业配置，参考 <a href="#">表12 DependJob数据结构说明</a> 。
concurrent	否	Integer	支持并发执行数量

表 5-123 Cron 数据结构说明

参数名	是否必选	参数类型	说明
startTime	是	String	调度开始时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。
endTime	否	String	调度结束时间，采用ISO 8601时间表示方法，格式为yyyy-MM-dd'T'HH:mm:ssZ，例如2018-10-22T23:59:59+08表示的时间为2018年10月22日23时59分59秒，在正8区，即北京时区。如果结束时间不配置，作业会按照调度周期一直执行下去。
expression	是	String	Cron表达式，格式为"<秒> <分> <时> <天> <月> <星期>"，每个字段允许的输入值参考 <a href="#">表5-125</a> 。
expressionTime Zone	否	String	Cron表达式对应的时区信息，例如GMT +8。 默认值：使用DataArts Studio服务端所在的时区。
dependPrePeriod	否	Boolean	是否依赖本作业上一个运行周期任务的执行结果。 默认值：false
dependJobs	否	DependJob数据结构	依赖其它作业配置，参考 <a href="#">表12 DependJob数据结构说明</a> 。

参数名	是否必选	参数类型	说明
intervalType	否	String	调度间隔类型: <ul style="list-style-type: none"> <li>• minutes: 分钟</li> <li>• hours: 小时</li> <li>• days: 天</li> <li>• weeks: 周</li> <li>• months: 月</li> <li>• seasons: 季</li> <li>• years: 年</li> </ul>

表 5-124 Event 数据结构说明

参数名	是否必选	参数类型	说明
eventType	是	String	选择对应的连接名称与topic, 当有新的kafka消息时将会触发作业运行一次 配置为: KAFKA 事件类型, 当前只支持监听DIS通道的新上报数据事件, 每上报一条数据, 触发作业运行一次。 配置为: DIS 选择要监听的OBS路径, 如果该路径下有新增文件, 则触发调度; 新增的文件的路径名, 可以通过变量Job.trigger.obsNewFiles引用。前提条件: 该OBS路径已经配置DIS消息通知。 配置为: OBS
channel	是	String	DIS通道名称。 通过DIS管理控制台获取通道名称: <ol style="list-style-type: none"> <li>1. 登录管理控制台。</li> <li>2. 单击“数据接入服务”, 左侧列表选择“通道管理”。</li> <li>3. 通道管理页面中列出了用户拥有的通道。</li> </ol>
failPolicy	否	String	执行失败处理策略: <ul style="list-style-type: none"> <li>• SUSPEND: 挂起</li> <li>• IGNORE: 忽略失败, 读取下一事件</li> </ul> 默认值: SUSPEND

参数名	是否必选	参数类型	说明
concurrent	否	int	调度并发数 取值范围: 1~128 默认值: 1
readPolicy	否	String	读取策略: <ul style="list-style-type: none"> <li>LAST: 从上次位置读取</li> <li>NEW- 从最新位置读取</li> </ul> 默认值: LAST

表 5-125 Cron 表达式字段取值

字段	取值范围	允许的特殊字符	说明
秒	0-59	, - * /	当前版本只允许输入0。
分	0-59	, - * /	-
时	0-23	, - * /	-
天	1-31	, - * ? / L W C	-
月	1-12	, - * /	当前版本只允许输入*。
星期	1-7	, - * ? / L C #	从星期日开始算起。

表 5-126 DependJob 数据结构说明

参数名	是否必选	参数类型	说明
jobs	是	List<String>	依赖的作业名称列表, 必须依赖已存在的作业。
dependPeriod	否	String	依赖周期: <ul style="list-style-type: none"> <li>SAME_PERIOD: 依赖被依赖作业的同周期任务的执行结果。</li> <li>PRE_PERIOD: 依赖被依赖作业的前一周期的任务的执行结果。</li> </ul> 默认值: SAME_PERIOD

参数名	是否必选	参数类型	说明
dependFailPolicy	否	String	依赖作业任务执行失败处理策略: <ul style="list-style-type: none"> <li>• FAIL: 停止作业, 设置作业为失败状态</li> <li>• IGNORE: 继续执行作业</li> <li>• SUSPEND: 挂起作业</li> </ul> 默认值: FAIL
sameWorkspaceJobs	否	DependWorkspaceJob数据结构	依赖本工作空间作业列表, 请参考表 5-145。
otherWorkspaceJobs	否	DependWorkspaceJob数据结构	依赖其他工作空间作业列表, 请参考表 5-145。

表 5-127 Property 属性说明

参数名	是否必选	参数类型	说明
name	是	String	属性名称
value	是	String	属性值

表 5-128 Hive SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 MRS Hive中的数据库, 默认default。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数, key、value形式, 多个参数间用\n分隔, 例如 key1=value1\nkey2=value2

表 5-129 Spark SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 MRS Spark SQL中的数据库，默认 default。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数，key、value形式，多个参数间用\n分隔，例如 key1=value1\nkey2=value2

表 5-130 DWS SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 DWS服务中的数据库，默认postgres。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数，key、value形式，多个参数间用\n分隔，例如key1=value1\nkey2=value2

表 5-131 DLI SQL 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
database	否	String	数据库名称 DLI服务中数据库。
connectionName	否	String	连接名称
scriptArgs	否	String	脚本参数，key、value形式，多个参数间用\n分隔，例如key1=value1\nkey2=value2

表 5-132 Shell 节点属性说明

参数名	是否必选	参数类型	说明
scriptName	是	String	脚本名称
connectionName	是	String	连接名称
arguments	否	String	Shell脚本参数

表 5-133 CDM Job 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	集群名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表可以获取到集群名称。
jobName	是	String	作业名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表单击对应的集群管理，进入后在作业管理中获取作业名称。

表 5-134 DISTransferTask 节点属性说明

参数名	是否必选	参数类型	说明
streamName	是	String	DIS通道名称 通过DIS管理控制台获取通道名称： 1. 登录管理控制台。 2. 单击“数据接入服务”，左侧列表选择“通道管理”。 3. 通道管理中列出了用户拥有的通道。
destinationType	是	String	转储目标： <ul style="list-style-type: none"> <li>CloudTable</li> <li>OBS</li> </ul>
duplicatePolicy	是	String	转储任务重名处理策略： <ul style="list-style-type: none"> <li>OVERWRITE：覆盖</li> <li>IGNORE：忽略</li> </ul>



参数名	是否必选	参数类型	说明
configuration	是	数据结构	转储配置，参考 <a href="#">添加转储任务</a> 的 obs_destination_descriptor 参数说明和 cloudtable_destination_descriptor 参数说明。

表 5-135 CloudTableManager 节点属性说明

参数名	是否必选	参数类型	说明
namespace	否	String	命名空间 默认值: default
action	是	String	动作类型: <ul style="list-style-type: none"> <li>CREATE_TABLE: 创建表</li> <li>DELETE_TABLE: 删除表</li> </ul>
table	否	String	表名
columnFamily	否	String	列族

表 5-136 OBSManager 节点属性说明

参数名	是否必选	参数类型	说明
action	是	String	动作类型: <ul style="list-style-type: none"> <li>CREATE_PATH: 创建OBS路径</li> <li>DELETE_PATH: 删除OBS路径</li> </ul>
path	是	String	OBS路径

表 5-137 RESTAPI 节点属性说明

参数名	是否必选	参数类型	说明
url	是	String	URL地址 云服务提供的URL地址。

参数名	是否必选	参数类型	说明
method	是	String	HTTP方法： <ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> </ul>
headers	否	String	HTTP消息头，每个消息头的格式为<消息头名称>=<值>，多个消息头之间使用换行符分割。
body	否	String	消息体

表 5-138 SMN 节点属性说明

参数名	是否必选	参数类型	说明
topic	是	String	SMN主题URN 通过SMN管理控制台获取SMN主题URN： <ol style="list-style-type: none"> <li>1. 登录管理控制台。</li> <li>2. 单击“消息通知服务”，左侧列表选择“主题管理 &gt; 主题”。</li> </ol> 在主题列表中可以获取到SMN主题URN。
subject	是	String	消息标题，给邮箱订阅者发送邮件时作为邮件主题。
messageType	是	String	消息类型 <ul style="list-style-type: none"> <li>• NORMAL</li> <li>• STRUCTURE</li> <li>• TEMPLATE</li> </ul>
message	是	String	发送的消息。

表 5-139 MRS Spark 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
resourcePath	是	String	自定义Spark Jar包OBS资源路径
parameters	是	String	Spark Jar包自定义参数 对于开发的自定义Jar包，可以在此处对参数进行输入替换
input	否	String	输入路径 MRS Spark作业输入数据路径，可以为HDFS或者是OBS路径。
output	否	String	输出路径 MRS Spark作业输出数据路径，可以为HDFS或者是OBS路径。
programParameter	否	String	运行程序参数 允许多个key:value，多个参数要用竖线隔开。

表 5-140 MapReduce 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
resourcePath	是	String	资源路径

参数名	是否必选	参数类型	说明
parameters	是	String	MapReduce作业参数
input	是	String	输入路径 MapReduce作业输入数据路径，可以为HDFS或者是OBS路径。
output	是	String	输出路径 MapReduce作业输出数据路径，可以为HDFS或者是OBS路径。

表 5-141 DLI Spark 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	DLI队列名称 通过DLI管理控制台获取队列名称： 1. 登录管理控制台。 2. 单击“数据湖探索服务”，单击“队列管理”。 在队列管理列表中可以获取到队列名称。
jobName	是	String	DLI作业名称 通过DLI管理控制台获取作业名称： 1. 登录管理控制台。 2. 单击“数据湖探索服务”，单击Spark作业。 3. 选择“作业管理”。 在作业管理列表中可以获取到作业名称。
resourceType	否	String	DLI作业运行资源类型，可选 1. OBS路径：OBS 2. DLI程序包：DLIResources
jobClass	否	String	主类名称。当应用程序类型为“.jar”时，主类名称不能为空。
resourcePath	是	String	JAR包资源路径
jarArgs	否	String	主类入口参数
sparkConfig	否	String	Spark作业运行参数

表 5-142 BasicConfig 作业基本信息说明

参数名	是否必选	参数类型	说明
owner	否	String	作业责任人，长度不能超过128个字符。
isIgnoreWaiting	否	integer	实例超时是否忽略等待时间，取值范围为0和1，0:表示实例超时不忽略等待时间1:表示实例超时忽略等待时间
priority	否	int	作业优先级，取值范围[0, 2]，默认值是0。0代表高优先级，1代表中优先级，2代表低优先级。
executeUser	否	String	作业执行用户，必须是已存在的用户名。
instanceTimeout	否	int	实例超时时间，单位是分钟，取值范围 [5, 1440]，默认值是60
customFields	否	Map<String,String>	用户自定义字段，长度不能超过2048个字符。
agency	否	String	代理名称
tags	否	List<String>	作业标签列表

表 5-143 MRS Flink 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获得到集群名称。
jobName	是	String	MRS作业名称 用户自定义。
flinkJobType	是	String	客户选择是Flink SQL作业或Flink JAR作业。

参数名	是否必选	参数类型	说明
flinkJobProcessType	是	String	客户选择批处理模式或流处理模式。
scriptName	否	String	客户选择Flink SQL关联的SQL脚本。
resourcePath	否	String	自定义Flink Jar包OBS资源路径
input	否	String	输入路径 MRS Flink作业输入数据路径，可以为HDFS或者是OBS路径。
output	否	String	输出路径 MRS Flink作业输出数据路径，可以为HDFS或者是OBS路径。
programParameter	否	String	运行程序参数 允许多个key:value，多个参数要用竖线隔开。

表 5-144 MRS HetuEngine 节点属性说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表单击“集群列表 > 现有集群”。 在现有列表中可以获取到集群名称。
jobName	是	String	MRS作业名称 用户自定义。

参数名	是否必选	参数类型	说明
statementOrScript	是	String	选择使用SQL语句和关联SQL脚本。
scriptName	否	String	选择关联的SQL脚本。
statement	否	String	自定义的SQL内容。
Data Warehouse	是	String	指定HetuEngine服务所需数据连接。
Schema	是	String	使用HetuEngine服务所要访问的数据源schema名称。
Database	是	String	使用HetuEngine服务所要访问的数据源database名称。
Queue	否	String	使用HetuEngine服务所需资源队列名称。

表 5-145 DependWorkSpaceJob 数据结构说明

参数名	是否必选	参数类型	说明
workSpace	是	String	工作空间名称。
workSpaceId	否	String	工作空间ID
jobName	是	String	作业名称。
dependOnLastPeriod	否	Boolean	是否依赖上游作业最近一个实例。
dependType	否	String	依赖的规则
dependOnLastNPeriodValue	否	Integer	依赖上N个周期

## 请求示例

查询作业myJob的详情。

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob
```

## 响应示例

### 📖 说明

调用查询作业详情接口时，如果在参数中带上dependencies=true，则会返回下游依赖当前作业的作业，只返回第一层。

GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/xxx\_1?dependencies=true

xxx\_1表示作业名。

响应消息会返回下游依赖当前作业的作业，如下示例中的响应结果消息字段downstreamJobs。

- 成功响应

```
{
  "basicConfig": {
    "agency": "",
    "customFields": {},
    "executeUser": "",
    "instanceTimeout": 0,
    "owner": "test_user",
    "priority": 0,
    "tags": {}
  },
  "cleanoverdueDays": "60",
  "cleanWaitingJob": "cleanup",
  "createTime": "1695642745000",
  "downstreamJobs": [
    {
      "jobId": "163240",
      "jobName": "xxx_2",
      "workSpace": "workspace_name"
    }
  ],
  "description": "",
  "emptyRunningJob": "0",
  "id": "608627",
  "lastUpdateUser": "test_user",
  "logPath": "obs://dlf-test-log",
  "name": "myJob",
  "nodes": [
    {
      "failPolicy": "FAIL_CHILD",
      "location": {
        "x": "-45.5",
        "y": "-134.5"
      },
      "maxExecutionTime": 360,
      "name": "MRS_Hive_SQL",
      "pollingInterval": 20,
      "preNodeName": [],
      "properties": [
        {
          "name": "scriptName",
          "value": "test_hive_sql"
        },
        {
          "name": "connectionName",
          "value": "mrs_hive_test"
        },
        {
          "name": "database",
          "value": "default"
        },
        {
          "name": "scriptArgs",
          "value": "test_var=111"
        }
      ]
    },
    {
      "retryInterval": 120,
      "retryTimes": 0,
      "type": "HiveSQL"
    }
  ]
}
```



```

    }
  ],
  "processType": "BATCH",
  "schedule": {
    "type": "CRON"
  }
}

```

- 失败响应

HTTP状态码 400

```

{
  "error_code": "DLF0100",
  "error_msg": "The job does not exists."
}

```

### 5.3.5 查询作业文件

#### 功能介绍

从OBS路径上导入作业前，可以通过此接口查询导入作业文件中是否有作业和脚本。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
POST /v1/{project\_id}/jobs/check-file
- 参数说明

表 5-146 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 5-147 参数说明

参数名	是否必选	参数类型	说明
path	是	String	有OBS场景：作业定义文件在OBS上的路径。例如obs://myBucket/jobs.zip。

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

## 响应参数

表 5-148 参数说明

参数名	是否必选	参数类型	说明
jobs	否	List<Job>	作业信息，参考 <a href="#">表5-149</a>
scripts	否	List<Script>	脚本信息， <a href="#">表5-150</a>

表 5-149 job 字段数据结构说明

参数名	是否必选	参数类型	说明
params	否	Map<String,String>	作业参数
name	是	String	作业名称
path	是	String	作业路径

表 5-150 Script 字段数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	脚本名称
path	是	String	脚本路径

## 请求示例

查询OBS上作业文件中的参数定义。作业定义文件在OBS上的路径为**obs://aaaaa/DLF\_myJob.zip**。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/check-file
{
```

```
"path": "obs://aaaaa/DLF_myJob.zip"  
}
```

## 响应示例

- 成功响应

```
{  
  "jobs": [  
    {  
      "name": "test",  
      "path": "/test",  
      "params": {  
        "ddd": "dddd"  
      }  
    },  
    {  
      "name": "test1",  
      "path": "/test",  
      "params": {  
        "ddd": "dddd"  
      }  
    }  
  ],  
  "scripts": [  
    {  
      "name": "script1",  
      "path": "/path1"  
    },  
    {  
      "name": "script2",  
      "path": "/path1"  
    }  
  ]  
}
```

- 失败响应

HTTP状态码 400

```
{  
  "error_code": "DLF0815",  
  "error_msg": "Fail to read OBS file."  
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.3.6 导出作业

### 功能介绍

导出zip压缩包格式的作业文件，包括作业定义、作业依赖的脚本、CDM作业定义。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v1/{project\_id}/jobs/{job\_name}/export

- 参数说明

表 5-151 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

## 请求参数

表 5-152 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

响应消息的Content-Type是application/octet-stream，需要解析转化为一个文件，可以参考[解析响应消息体中的流样例代码](#)。响应是一个压缩文件，文件名称格式DLF\_{job\_name}.zip，压缩文件里面的文件目录如下：

```
jobs
├─{job_name}.job
scripts
├─{script_name}.script
resources
├─{resource_name}.resource
```

上述文件名中的参数说明如[表5-153](#)所示。

表 5-153 参数说明

参数名	是否必选	参数类型	说明
job_name	是	String	作业名称
script_name	否	String	作业所依赖的脚本名称。
resource_name	否	String	作业所依赖的资源名称。

- {job\_name}.job文件内容  
文件中的参数与创建作业接口的请求参数一样，请参考[创建作业](#)。
- {script\_name}.script文件内容  
文件中的参数与创建资源接口的请求参数一样，请参考[创建脚本](#)。

## 请求示例

导出作业myJob。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob/export
```

## 响应示例

- 成功响应  
HTTP状态码 200  
导出文件名为DLF\_myJob.zip，解压后文件结构如下：

```
jobs
├─myJob.job
scripts
├─CS_PROCESS_TRIP.script
├─TRIP_RAW_STANDARD.script
```

- 失败响应  
HTTP状态码 400  

```
{
  "error_code": "DLF0100",
  "error_msg": "The job does not exists."
}
```

## 5.3.7 批量导出作业

### 功能介绍

此接口可以批量导出DLF作业定义，包括作业依赖的脚本和CDM作业定义。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v1/{project\_id}/jobs/batch-export
- 参数说明

表 5-154 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 5-155 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

表 5-156 参数说明

参数名	是否必选	参数类型	说明
jobList	是	列表数据结构	要导出的作业名称列表，一次最多支持导出100个作业。
exportDepend	否	boolean	是否导出作业依赖的脚本和资源。 默认值: true

## 响应参数

批量导出作业响应消息和导出作业响应消息一致，请参考 [响应参数](#)。

## 请求示例

导出两个作业，作业名称分别为job\_batch和job\_stream。是否导出作业依赖的脚本和资源默认为true。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/batch-export
{
  "jobList":["job_batch","job_stream"],
  "exportDepend":true
}
```

## 响应示例

- 成功响应

响应消息的Content-Type是application/octet-stream，需要解析转化为一个文件，可以参考[解析响应消息体中的流样例代码](#)。响应是一个压缩文件，文件名为jobs.zip，解压后文件结构如下：

```
jobs
+---job_batch
|   dws_sql.script
|   job_batch.job
+---job_stream
job_stream.job
```

job\_batch.job和job\_stream.job是作业定义文件

dws\_sql.script是job\_batch.job所使用的DWS SQL脚本文件。

- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.3051",
  "error_msg": "The request parameter is invalid."
}
```

## 5.3.8 导入作业

### 功能介绍

将存放在OBS桶的作业文件导入到DLF，一次可以导入一个或多个作业。

#### 说明

执行本接口之前，需要将作业文件放到OBS的桶中。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v1/{project\_id}/jobs/import
- 参数说明

表 5-157 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 5-158 参数说明

参数名	是否必选	参数类型	说明
path	是	String	有OBS场景：作业定义文件在OBS上的路径，作业定义文件的格式参考导出作业的响应消息。例如obs://myBucket/jobs.zip。
params	否	Map<String,String>	公共作业参数。

参数名	是否必选	参数类型	说明
sameNamePolicy	否	String	重名策略 <ul style="list-style-type: none"> <li>• SKIP: 跳过</li> <li>• OVERWRITE: 覆盖</li> </ul> 默认值是SKIP。
jobsParam	否	List<JobParam>	指定作业参数，参考 <a href="#">表5-159</a> 。
executeUser	否	String	执行作业的用户。
targetStatus	否	String	在开启审批开关后，需要填写该字段。表示创建作业的目标状态，有三种状态：SAVED、SUBMITTED和PRODUCTION，分别表示作业创建后是保存态，提交态，生产态。 <ul style="list-style-type: none"> <li>• 保存态表示作业仅保存，无法调度运行，需要提交并审核通过后才能运行。</li> <li>• 提交态表示作业保存后会自动提交，需要审核通过才能运行。</li> <li>• 生产态表示作业跳过审批环节，创建后可以直接运行。注意：只有工作空间的管理员用户才能创建生产态的作业。</li> </ul>
approvers	否	List<JobApprover>	在开启审批开关后，需要填写该字段，表示作业审批人，参考 <a href="#">表5-160</a> 。审批开关的配置请参见 <a href="#">审批中心</a> 。
resources	否	List<JobResourceInfo>	如需替换资源，需要填写该字段，包含替换的资源名和资源类型和替换后的资源名。参考 <a href="#">表5-161</a> 。

表 5-159 JobParam 属性说明

参数名	是否必选	参数类型	说明
name	是	String	作业名称。
params	否	Map<String,String>	作业参数。



表 5-160 审批人属性说明

参数名	是否必选	参数类型	说明
approverName	是	String	审批人名称。

表 5-161 资源表说明

参数名	是否必选	参数类型	说明
name	是	String	资源名。
type	是	String	资源类型，支持的资源类型如下： DWS_CONNECTION、 DIS_STREAM、 CDM_CLUSTER、 MRS_CLUSTER、 MRS_HIVE_CONNECTION、 MRS_SPARK_CONNECTION、 GES_GRAPH、 ROMA_INSTANCE、 TICS_INSTANCE、 DRS_TASK。
replace	否	String	替换后的资源名。

## 响应参数

表 5-162 参数说明

参数名	是否必选	参数类型	说明
taskId	是	String	任务ID，请调用查询系统任务接口获取导入状态。

## 请求示例

从OBS路径上导入作业，如果有重名作业、脚本，则覆盖。所导入的作业定义文件在OBS上的路径为**obs://aaaaa/job\_batch.zip**，指定的作业参数的作业名称为**job\_batch**。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/import
{
  "path": "obs://aaaaa/job_batch.zip",
  "jobsParam": [
```

```
{
  "name": "job_batch"
}
]
```

打开审批开关的情况下，从OBS路径上导入作业，如果有重名作业、脚本，则覆盖。

POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/import

```
{
  "path": "obs://aaaaa/job_batch.zip",
  "jobsParam": [
    {
      "name": "job_batch"
    }
  ],
  "targetStatus": "SUBMITTED",
  "approvers": [
    {
      "approverName": "userName1"
    },
    {
      "approverName": "userName2"
    }
  ]
}
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "taskId": "008aae2e675933c7016759418e870000"
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.0815",
  "error_msg": "Fail to read OBS file."
}
```

## 5.3.9 立即执行作业

### 功能介绍

本接口用于立即执行作业，验证作业是否能执行成功。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v1/{project\_id}/jobs/{job\_name}/run-immediate
- 参数说明

表 5-163 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

## 请求参数

表 5-164 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-165 参数说明

参数名	是否必选	参数类型	说明
jobParams	否	List<JobParam>	作业立即执行的参数。

表 5-166 JobParam 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	参数名称，长度不能超过64个字符。
value	是	String	参数值，长度不能超过1024个字符。
type	否	String	参数类型。 <ul style="list-style-type: none"> <li>variable: 变量</li> <li>constants: 常量</li> </ul>

## 响应参数

表 5-167 参数说明

参数名	是否必选	参数类型	说明
instanceId	是	Long	作业实例id。

## 请求示例

单次执行作业myJob。作业中的参数为aaa，参数值为111，作业中的参数为bbb，参数值为222。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob/run-immediate
{
  "jobParams":[
    {
      "name":"aaa",
      "value":"111"
    },
    {
      "name":"bbb",
      "value":"222"
    }
  ]
}
```

## 响应示例

- 成功响应

```
{
  "instanceId":132343
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code":"DLF.0100",
  "error_msg":"The job does not exists."
}
```

## 5.3.10 启动作业

### 功能介绍

本接口用于执行调度一个作业。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式

```
POST /v1/{project_id}/jobs/{job_name}/start
```

- 参数说明

表 5-168 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

## 请求参数

表 5-169 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-170 参数说明

参数名	是否必选	类型	描述
jobParams	否	List<JobParam>	作业启动的参数。
start_date	否	Long	启动日期。比如：20241030
ignore_first_self_dep	否	boolean	忽略首天依赖。

表 5-171 JobParam 数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	参数名称，长度不能超过64个字符。
value	是	String	参数值，长度不能超过1024个字符。
paramType	否	String	参数类型。 <ul style="list-style-type: none"> <li>• variable: 变量</li> <li>• constants: 常量</li> </ul>

## 响应参数

无

## 请求示例

启动作业myJob。

POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob/start

```
{
  "jobParams": [
    {
      "name": "string",
      "value": "string",
      "paramType": "variable"
    }
  ]
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.0100",
  "error_msg": "The job does not exists."
}
```

## 5.3.11 停止作业

### 功能介绍

本接口用于停止调度一个作业。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v1/{project\_id}/jobs/{job\_name}/stop
- 参数说明

表 5-172 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

## 请求参数

表 5-173 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"><li>如果不设置该参数，默认查询default工作空间下的数据。</li><li>如果需要查询其他工作空间的数据，需要带上该消息头。</li></ul>

## 响应参数

无

## 请求示例

停止作业myJob。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob/stop
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{  
  "error_code": "DLF.0100",  
  "error_msg": "The job does not exists."  
}
```

## 5.3.12 删除作业

### 功能介绍

本接口用于删除DLF作业，调度中的作业不允许删除，若要删除调度中的作业，请先停止该作业。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
DELETE /v1/{project\_id}/jobs/{job\_name}
- 参数说明

表 5-174 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

## 请求参数

表 5-175 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-176 参数说明

参数名	是否必选	参数类型	说明
approvers	否	List<JobApprover>	在开启审批开关后，需要填写该字段，表示作业审批人，参考 <a href="#">表5-177</a> 。 审批开关的配置请参见 <a href="#">审批中心</a> 。

表 5-177 审批人属性说明

参数名	是否必选	参数类型	说明
approverName	是	String	审批人名称。

## 响应参数

无

## 请求示例

```
DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/joba
```

打开审批开关的情况下，删除作业的示例。

```
DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/joba
{
```



```
"approvers": [  
  {  
    "approverName": "userName1"  
  },  
  {  
    "approverName": "userName2"  
  }  
]
```

## 响应示例

- 响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{  
  "error_code": "DLF0100",  
  "error_msg": "The job does not exists."  
}
```

## 状态码

状态码请参见[状态码](#)。

### 5.3.13 停止作业实例

#### 功能介绍

停止指定作业实例。只有作业实例的状态是运行中，才能停止作业实例。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
POST /v1/{project\_id}/jobs/{job\_name}/instances/{instance\_id}/stop
- 参数说明

表 5-178 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。
instance_id	是	Long	作业实例ID，获取方法请参见 <a href="#">查询作业实例列表</a>

## 请求参数

表 5-179 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

## 响应参数

无

## 请求示例

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/job_batch/instances/34765/stop
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.0137",
  "error_msg": "Job instance does not exist."
}
```

## 5.3.14 重跑作业实例

### 功能介绍

重跑指定作业实例。只有作业实例的状态是成功、失败和取消，才能重跑作业实例。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v1/{project\_id}/jobs/{job\_name}/instances/{instance\_id}/restart
- 参数说明

表 5-180 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。
instance_id	是	Long	作业实例ID，获取方法请参见 <a href="#">查询作业实例列表</a> 。

## 请求参数

表 5-181 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-182 参数说明

参数名	是否必选	类型	描述
retry_location	否	String	作业重跑的位置。error_node（从错误节点开始重跑）first_node（从第一个节点开始重跑）、specified_node（从指定的节点开始），默认从重跑位置默认节点开始。
job_parameter_version	否	String	使用的作业参数。original_version（使用原有作业参数）、latest_version（使用最新提交作业参数重跑）。
ignore_obs_monitor	否	boolean	是否忽略obs监听。
node_name	否	String	指定的节点名。当retry_location选择specified_node指定重跑节点。

## 响应参数

无

## 请求示例

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/job_batch/instances/34765/restart
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{  
  "error_code": "DLF.0137",  
  "error_msg": "Job instance does not exist."  
}
```

## 5.3.15 查询实时作业运行状态

### 功能介绍

查询指定实时作业的运行状态。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
GET /v1/{project\_id}/jobs/{job\_name}/status
- 参数说明

表 5-183 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

## 请求参数

表 5-184 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

## 响应参数

表 5-185 参数说明

参数名	是否必选	参数类型	说明
name	是	String	解决方案名称。
nodes	否	列表数据结构	节点状态列表。
status	否	String	作业状态： • STARTING: 启动中 • NORMAL: 正常 • EXCEPTION: 异常 • STOPPING: 停止中 • STOPPED: 停止
startTime	是	Date	启动时间。
endTime	否	Date	停止时间。
lastUpdateTime	否	Date	状态最后更新时间。

表 5-186 nodes 字段数据结构说明

参数名	是否必选	参数类型	说明
name	是	String	节点名称。

参数名	是否必选	参数类型	说明
status	否	String	节点状态: <ul style="list-style-type: none"> <li>STARTING: 启动中</li> <li>NORMAL: 正常</li> <li>EXCEPTION: 异常</li> <li>STOPPING: 停止中</li> <li>STOPPED: 停止</li> </ul>
logPath	否	String	节点运行日志路径
type	是	String	节点类型: <ul style="list-style-type: none"> <li>Hive SQL: 执行Hive SQL脚本</li> <li>Spark SQL: 执行Spark SQL脚本</li> <li>DWS SQL: 执行DWS SQL脚本</li> <li>DLI SQL: 执行DLI SQL脚本</li> <li>Shell: 执行Shell SQL脚本</li> <li>CDM Job: 执行CDM作业</li> <li>DISTransferTask: 创建DIS转储任务</li> <li>CS Job: 创建CloudStream作业, 并启动作业</li> <li>CloudTableManager节点: CloudTable表管理, 参考创建和删除表。</li> <li>OBSManager: OBS路径管理, 包括创建和删除路径</li> <li>RESTAPI: REST API请求</li> <li>SMN: 发送短信或邮件</li> <li>MRS Spark: 执行MRS服务的Spark作业</li> <li>MapReduce: 执行MRS服务的MapReduce作业</li> </ul>

## 请求示例

查询实时作业job\_sms作业运行状态以及各个节点运行状态。  
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/job\_sms/status

## 响应示例

- 成功响应

```
{
  "name": "job_sms",
  "nodes": [
    {
      "bufferRecords": 0,
      "jobInstanceld": 0,
    }
  ]
}
```

```

        "LastInstanceStatus": "waiting",
        "name": "MRS_Flink_Job_8635",
        "speed": 0,
        "totalGetBytes": 0,
        "totalGetRecords": 0,
        "totalPutBytes": 0,
        "totalPutRecords": 0
    }
],
    "status": "NORMAL"
}

```

- 失败响应

HTTP状态码 400

```

{
    "error_code": "DLF.0100",
    "error_msg": "The job does not exists."
}

```

## 5.3.16 查询作业实例列表

### 功能介绍

查询作业实例列表。

对于配置了周期调度或事件触发调度的批处理作业，触发一次运行就会生产一个作业实例。如果实时作业里面包含了配置周期调度或事件触发调度节点，可以通过此接口查询节点所关联的子作业的实例列表，参数jobName格式为[实时作业名称]\_[节点名称]。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式

GET /v1/{project\_id}/jobs/instances/detail?

jobName={jobName}&minPlanTime={minPlanTime}&maxPlanTime={maxPlanTime}&limit={limit}&offset={offset}&status={status}&instanceType={instanceType}&preciseQuery={preciseQuery}

- 参数说明

表 5-187 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

参数名	是否必选	参数类型	说明
jobName	否	String	作业名称。 <ul style="list-style-type: none"> <li>如果要查询指定批处理作业的实例列表, jobName就是批处理作业名称;</li> <li>如果要查询实时作业下某个节点关联的子作业, jobName格式为[实时作业名称]_[节点名称]。</li> </ul>
minPlanTime	否	Long	返回作业实例开始时间大于minPlanTime的作业实例, 单位为毫秒ms, 默认设置为查询当天0点。
maxPlanTime	否	Long	返回作业实例开始时间小于maxPlanTime的作业实例, 单位为毫秒ms。
limit	否	int	分页返回结果, 指定每页最大记录数。 范围[1,1000] 默认值: 10
offset	否	int	分页的起始页, 默认值为0。取值范围大于等于0。
status	否	String	实例运行状态: <ul style="list-style-type: none"> <li>waiting: 等待运行</li> <li>running: 运行中</li> <li>success: 运行成功</li> <li>fail: 运行失败</li> <li>running-exception: 运行异常</li> <li>pause: 暂停</li> <li>manual-stop: 取消</li> </ul>
instanceType	否	int	作业调度方式: <ul style="list-style-type: none"> <li>0: 正常调度</li> <li>2: 手工调度</li> <li>5: 补数据</li> <li>6: 子作业调度</li> <li>7: 单次调度</li> </ul>
preciseQuery	否	boolean	支持通过作业名进行精确查询。



## 请求参数

表 5-188 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

## 响应参数

表 5-189 参数说明

参数名	是否必选	参数类型	说明
total	是	int	总记录数
instances	是	List<Instance>	作业实例状态，参考 <a href="#">表5-190</a> 。

表 5-190 Instance 属性说明

参数名	是否必选	参数类型	说明
jobName	是	String	作业名称。如果要查询指定批处理作业的实例列表，jobName就是批处理作业名称；如果要查询实时作业下某个节点关联的子作业，jobName格式为[实时作业名称]_[节点名称]。
jobId	否	Long	作业id。
jobInstanceName	是	String	作业实例运行时日志记录的实例名称，非作业定义的名称。

参数名	是否必选	参数类型	说明
status	是	String	状态： <ul style="list-style-type: none"> <li>• waiting：等待运行</li> <li>• running：运行中</li> <li>• success：运行成功</li> <li>• fail：运行失败</li> <li>• running-exception：运行异常</li> <li>• pause：暂停</li> <li>• manual-stop：取消</li> </ul>
planTime	是	Long	作业实例计划执行时间
startTime	是	Long	作业实例实际执行开始时间
endTime	否	Long	作业实例实际执行结束时间
executeTime	否	Long	执行耗时，单位：毫秒
instanceId	是	Long	作业实例ID，用于查询作业实例详情。
submitTime	是	Long	作业提交运行时间
instanceType	是	int	作业调度方式： <ul style="list-style-type: none"> <li>• 0：正常调度</li> <li>• 2：手工调度</li> <li>• 5：补数据</li> <li>• 6：子作业调度</li> <li>• 7：单次调度</li> </ul>
forceSuccess	否	boolean	作业实例状态筛选为强制成功 默认值：false
ignoreSuccess	否	boolean	作业实例状态筛选为忽略失败 默认值：false
version	否	int	作业实例版本。

## 请求示例

查询批处理作业job\_batch的实例列表

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/instances/detail?jobName=job_batch
```

## 响应示例

成功响应

```
{
  "total": 2,
  "instances": [{
```

```

"endTime": 1551671598000,
"executeTime": 0.3,
"forceSuccess": false,
"ignoreSuccess": false,
"instanceId": 34765,
"instanceType": 2,
"jobId": 608627,
"jobName": "job_batch",
"jobInstanceName": "job_batch",
"planTime": 1551671580000,
"startTime": 1551671580000,
"status": "success",
"submitTime": 1550910278706,
"version": 1
},
{
"endTime": 1551671538000,
"executeTime": 0.3,
"forceSuccess": false,
"ignoreSuccess": false,
"instanceId": 34764,
"instanceType": 5,
"jobId": 608627,
"jobName": "job_batch",
"jobInstanceName": "job_batch",
"planTime": 1551671520000,
"startTime": 1551671521000,
"status": "success",
"submitTime": 1550910278706,
"version": 1
}
}

```

### 5.3.17 查询作业实例详情

#### 功能介绍

查询指定作业实例详情信息，包括作业实例内各个节点的执行信息。

#### 调试

您可以在 [API Explorer](#) 中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
GET /v1/{project\_id}/jobs/{job\_name}/instances/{instance\_id}
- 参数说明

表 5-191 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

参数名	是否必选	参数类型	说明
instance_id	是	string	作业实例ID, 获取方法请参见 <a href="#">查询作业实例列表</a> 的响应参数。

## 请求参数

表 5-192 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数, 默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据, 需要带上该消息头。</li> </ul>

## 响应参数

表 5-193 参数说明

参数名	是否必选	参数类型	说明
jobName	是	String	作业名称。
instanceId	是	Long	作业实例ID。
status	是	String	作业实例状态: <ul style="list-style-type: none"> <li>waiting: 等待运行</li> <li>running: 运行中</li> <li>success: 运行成功</li> <li>fail: 运行失败</li> <li>running-exception: 运行异常</li> <li>pause: 暂停</li> <li>manual-stop: 取消</li> </ul>
planTime	是	Long	作业实例计划执行时间。
startTime	是	Long	作业实例实际执行开始时间。
endTime	否	Long	作业实例实际执行结束时间。
executeTime	否	Long	执行耗时, 单位: 毫秒。
total	是	Integer	总的节点数据条数。

参数名	是否必选	参数类型	说明
nodes	是	List<Node>	节点实例状态，参考 <a href="#">表5-194</a>
instanceType	是	Integer	作业调度方式： <ul style="list-style-type: none"> <li>0：正常调度</li> <li>2：手工调度</li> <li>5：补数据</li> <li>6：子作业调度</li> <li>7：单次调度</li> </ul>
forceSuccess	否	boolean	作业实例状态筛选为强制成功。 默认值：false
ignoreSuccess	否	boolean	作业实例状态筛选为忽略失败。 默认值：false

表 5-194 Node 属性说明

参数名	是否必选	参数类型	说明
nodeName	是	String	节点名称。
status	是	String	节点状态： <ul style="list-style-type: none"> <li>waiting：等待运行</li> <li>running：运行中</li> <li>success：运行成功</li> <li>fail：运行失败</li> <li>skip：跳过</li> <li>pause：暂停</li> <li>manual-stop：取消</li> </ul>
queue	是	String	DLI资源队列名称。 <b>说明</b> 在返回响应中，仅DLI SQL或者DLI SPARK算子会返回DLI队列名称。
planTime	是	Long	作业实例计划执行时间。
startTime	是	Long	节点实际执行开始时间。
endTime	否	Long	节点实际执行结束时间。

参数名	是否必选	参数类型	说明
type	是	String	节点类型： <ul style="list-style-type: none"> <li>• HiveSQL：执行Hive SQL脚本</li> <li>• SparkSQL：执行Spark SQL脚本</li> <li>• DWSSQL：执行DWS SQL脚本</li> <li>• DLISQL：执行DLI SQL脚本</li> <li>• Shell：执行Shell SQL脚本</li> <li>• CDMJob：执行CDM作业</li> <li>• DISTransferTask：创建DIS转储任务</li> <li>• CloudTableManager：CloudTable表管理，创建和删除表。</li> <li>• OBSManager：OBS路径管理，包括创建和删除路径。</li> <li>• RestClient：REST API请求</li> <li>• SMN：发送短信或邮件</li> <li>• MRSSpark：执行MRS服务的Spark作业</li> <li>• MapReduce：执行MRS服务的MapReduce作业</li> <li>• MRSFlinkJob：执行MRS服务的FlinkJob作业。</li> <li>• MRSHetuEngine：执行MRS服务的HetuEngine作业。</li> <li>• DLISpark：执行DLF服务的Spark作业</li> <li>• RDSSQL：传递SQL语句到RDS中执行。</li> <li>• ModelArts Train：执行ModelArts服务的workflow作业。</li> </ul>
retryTimes	否	Integer	失败重试次数。
instanceId	是	Long	作业实例id。
inputRowCount	否	Long	写入数据行数。
speed	否	number	写入速度(行/秒)。
logPath	否	String	节点执行的日志路径。

## 请求示例

查询作业job\_batch实例ID为34765的实例详情。

GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/job\_batch/instances/34765

## 响应示例

- 成功响应

```
{
  "jobName": "job_batch",
  "instanceId": 34765,
  "instanceType": 2,
  "status": "fail",
  "planTime": 1551425326540,
  "startTime": 1551425327000,
  "endTime": 1551425387000,
  "executeTime": 1,
  "forceSuccess": false,
  "ignoreSuccess": false,
  "total": 2,
  "nodes": [
    {
      "endTime": 1551671590000,
      "inputRowCount": 0,
      "instanceId": 34765,
      "nodeName": "Dummy_8556",
      "queue": "dlf_notdelete",
      "planTime": 1551671580000,
      "retryTimes": 0,
      "startTime": 1551671584000,
      "status": "success",
      "type": "Dummy"
    },
    {
      "endTime": 1551671598000,
      "inputRowCount": 0,
      "instanceId": 34765,
      "logPath": "obs://dlf-test-log/job_batch/2019-03-04 11_53_00.000/error/error.job",
      "nodeName": "error",
      "planTime": 1551671580000,
      "retryTimes": 0,
      "startTime": 1551671594000,
      "status": "success",
      "type": "DWS SQL"
    }
  ]
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.0137",
  "error_msg": "Job instance does not exist."
}
```

### 5.3.18 查询系统任务详情

#### 功能介绍

本接口用于查询异步任务详情。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式

GET /v1/{project\_id}/system-tasks/{task\_id}

- 参数说明

表 5-195 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
task_id	是	String	任务ID，获取方法请参见 <a href="#">导入作业</a> 的响应参数。

## 请求参数

表 5-196 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

表 5-197 参数说明

参数名	是否必选	参数类型	说明
id	是	String	任务ID。
name	是	String	任务名称。
startTime	是	Long	开始时间。
endTime	否	Long	结束时间。
lastUpdate	是	Long	更新时间。
status	是	String	状态： <ul style="list-style-type: none"> <li>• RUNNING</li> <li>• SUCCESSFUL</li> <li>• FAILED</li> </ul>
projectId	否	String	项目编号。



参数名	是否必选	参数类型	说明
subtasks	否	List<SubTask>	子任务，参考 <a href="#">表5-198</a> 。

表 5-198 SubTask 参数说明

参数名	是否必选	参数类型	说明
id	是	String	子任务ID。
name	是	String	子任务名称。
lastUpdate	是	Long	更新时间。
status	是	String	状态： <ul style="list-style-type: none"> <li>● RUNNING</li> <li>● SUCCESSFUL</li> <li>● FAILED</li> </ul>
taskId	否	String	任务消息。

## 请求示例

查询导出作业的任务。

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/systemTasks/008aae2e675d3bcb01675d40ab3f0005
```

## 响应示例

- 成功响应

```
{
  "endTime": 1544777083000,
  "id": "008aae2e67abe2ff0167abe3a28f0000",
  "lastUpdate": 1544777083000,
  "name": "从路径[obs://aaaaa/job_batch.zip]导入作业。",
  "projectId": "b384b9e9ab9b4ee8994c8633aabc9505",
  "startTime": 1544777081000,
  "status": "SUCCESSFUL",
  "subtasks": [
    {
      "id": "008aae2e67abe2ff0167abe3a7e70002",
      "lastUpdate": 1544777083000,
      "name": "导入作业[job_batch]。",
      "status": "SUCCESSFUL",
      "taskId": "008aae2e675d3bcb01675d40ab3f0005"
    }
  ]
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.20810",
```

```
"error_msg":"Task[008aae2e675d3bcb01675d40ab3f0005] does not exist"
}
```

## 5.4 连接管理 API (待下线)

### 5.4.1 创建连接 (待下线)

#### 说明

连接管理能力由管理中心统一提供，相关接口已不再维护，推荐使用管理中心API进行连接管理。

#### 功能介绍

创建一个新连接，当前支持的连接类型有DWS、DLI、SparkSQL、HIVE、RDS、CloudTable。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
POST /v1/{project\_id}/connections
- 参数说明

表 5-199 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 5-200 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-201 Connection 参数说明

参数名	是否必选	参数类型	说明
name	是	String	连接名称，只能包含四种字符：英文字母、数字、中划线和下划线，且长度小于等于100个字符。连接名称不能重复。
type	是	String	连接类型，包含： <ul style="list-style-type: none"> <li>• DWS</li> <li>• DLI</li> <li>• SparkSQL</li> <li>• HIVE</li> <li>• RDS</li> <li>• CloudTable</li> <li>• HOST</li> </ul>
config	否	Map<String,String>	连接的配置项，不同类型的连接配置项不同。DLI类型的连接不需要配置。config参数，其他类型的连接请参考连接配置项说明。
description	否	String	描述，长度不能超过255个字符。

表 5-202 DWS 类型的连接配置项说明

参数名	是否必选	参数类型	说明
clusterName	否	String	DWS集群名称。 通过DWS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“数据仓库服务”，左侧列表选择集群管理。 在集群管理列表中可以获取到集群名称。 默认值为空。

参数名	是否必选	参数类型	说明
ip	否	String	DWS集群的访问地址。 通过DWS管理控制台获取访问地址： 1. 登录管理控制台。 2. 单击“数据仓库服务”，左侧列表选择集群管理。 3. 单击集群名称，进入集群基本信息页面。 在数据库属性标签中可以查看到内网IP，如果有多个IP地址，选择第一个IP。 默认值为空。
port	否	String	DWS集群的访问端口。 通过DWS管理控制台获取访问集群的访问端口： 1. 登录管理控制台。 2. 单击“数据仓库服务”，左侧列表选择集群管理。 3. 单击集群名称，进入集群基本信息页面。 在数据库属性标签中可以获取到端口信息。例如8000，默认值为空。
userName	是	String	数据库的用户名，创建DWS集群的时输入的用户名。
password	是	String	数据库的访问密码，创建DWS集群的时候输入的密码。
sslEnable	是	boolean	是否使用ssl连接。
kmsKey	是	String	KMS密钥名称。 通过KMS管理控制台获取密钥名称： 1. 登录管理控制台。 2. 单击“密钥管理服务”，左侧列表选择密钥管理。 在密钥列表可以获取到密钥名称。
agentName	是	String	连接代理，即CDM集群名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表可以获取到集群名称。

表 5-203 Spark SQL 类型的连接配置项说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称。 通过MRS管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“MapReduce服务”，左侧列表选择现有集群。 在现有集群中可以获取到集群名称，例如mrsCluster1。
connectionMethod	是	String	连接方式： <ul style="list-style-type: none"> <li>agent：通过代理连接。</li> <li>direct：直接连接。</li> </ul>
userName	否	String	MRS集群的用户名，通过代理连接的时候，是必选项。
password	否	String	MRS集群的访问密码，通过代理连接的时候，是必选项。
agentName	否	String	连接代理，即CDM集群名称。通过代理连接的时候，是必选项。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表可以获取到集群名称。
kmsKey	否	String	KMS密钥名称。通过代理连接的时候，是必选项。 通过KMS管理控制台获取密钥名称： 1. 登录管理控制台。 2. 单击“密钥管理服务”，左侧列表选择密钥管理。 在密钥列表可以获取到密钥名称。

表 5-204 Hive 类型的连接配置项说明

参数名	是否必选	参数类型	说明
clusterName	是	String	MRS集群名称，例如mrsCluster1
connectionMethod	是	String	连接方式： <ul style="list-style-type: none"> <li>agent：通过代理连接。</li> <li>direct：直接连接。</li> </ul>

参数名	是否必选	参数类型	说明
userName	否	String	MRS集群的用户名，通过代理连接的时候，是必选项。
password	否	String	MRS集群的访问密码，通过代理连接的时候，是必选。
agentName	否	String	连接代理，即CDM集群名称。通过代理连接的时候，是必选项。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表可以获取到集群名称。
kmsKey	否	String	KMS密钥名称。通过代理连接的时候，是必选项。 通过KMS管理控制台获取密钥名称： 1. 登录管理控制台。 2. 单击“密钥管理服务”，左侧列表选择密钥管理。 在密钥列表可以获取到密钥名称。

表 5-205 RDS 类型的连接配置项说明

参数名	是否必选	参数类型	说明
ip	是	String	RDS的访问地址。 通过RDS管理控制台获取访问地址： 1. 登录管理控制台。 2. 单击“关系型数据库服务”，从左侧列表选择实例管理。 3. 单击某一个实例名称，进入实例基本信息页面。 在连接信息标签中可以获取到内网地址。
port	是	String	RDS的访问端口。 通过RDS管理控制台获取访问端口： 1. 登录管理控制台。 2. 单击“关系型数据库服务”，左侧列表选择实例管理。 3. 单击某一个实例名称，进入实例基本信息页面。 在连接信息标签中可以获取到数据库端口。

参数名	是否必选	参数类型	说明
userName	是	String	数据库的用户名，创建集群的时候，输入的用户名。
password	是	String	数据库的访问密码，创建集群的时候，输入的密码。
kmsKey	是	String	KMS密钥名称。 通过KMS管理控制台获取密钥名称： 1. 登录管理控制台。 2. 单击“密钥管理服务”，左侧列表选择密钥管理。 在密钥列表可以获取到密钥名称。
agentName	是	String	连接代理，即CDM集群名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表可以获取到集群名称。
driverName	是	String	驱动程序名称： <ul style="list-style-type: none"> <li>com.mysql.jdbc.Driver</li> <li>org.postgresql.Driver</li> </ul>
driverPath	是	String	驱动在OBS上路径。

表 5-206 CloudTable 类型的连接配置项说明

参数名	是否必选	参数类型	说明
clusterName	是	String	CloudTable集群名称 通过CloudTable管理控制台获取集群名称： 1. 登录管理控制台。 2. 单击“表格存储服务”，左侧列表选择集群模式。 在集群模式中获取到集群名称。

表 5-207 HOST 类型的连接配置项说明

参数名	是否必选	参数类型	说明
ip	是	String	主机的地址

参数名	是否必选	参数类型	说明
port	是	String	主机的SSH端口号
userName	是	String	主机的登录用户名
password	是	String	主机的登录方式为密码时，填写主机的登录密码。
agentName	是	String	连接代理，即CDM集群名称。 通过DataArts Studio管理控制台 > 数据集成，在批量数据迁移集群列表可以获取到集群名称。
kmsKey	是	String	KMS密钥名称。 通过KMS管理控制台获取密钥名称： 1. 登录管理控制台。 2. 单击“密钥管理服务”，左侧列表选择密钥管理。 在密钥列表可以获取到密钥名称。

## 响应参数

无

## 请求示例

创建一个连接。

POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections

```
{
  "name": "connection1",
  "type": "DWS",
  "config": {
    "clusterName": "test",
    "userName": "dbadmin",
    "password": "*****",
    "kmsKey": "cdm-dlf",
    "agentName": "cdm-donotdelete",
    "sslEnable": false
  }
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.6309",
  "error_msg": "The name already exists."
}
```



## 状态码

状态码请参见[状态码](#)。

## 5.4.2 查询连接列表（待下线）

### 说明

连接管理能力由管理中心统一提供，相关接口已不再维护，推荐使用管理中心API进行连接管理。

## 功能介绍

查询连接列表。

## 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

## URI

- URI格式  
GET /v1/{project\_id}/connections?  
offset={offset}&limit={limit}&connectionName={connectionName}
- 参数说明

表 5-208 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
offset	否	Integer	偏移量，取值范围大于等于0，默认值为0。
limit	否	Integer	分页返回结果，指定每页最大记录数。范围[1,100] 默认值：10
connectionName	否	String	连接名称

## 请求参数

表 5-209 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

## 响应参数

参数名	是否必选	参数类型	说明
total	是	Integer	总的连接个数
connections	是	List<Connections>	连接列表

表 5-210 connections 参数说明

参数名	是否必选	参数类型	说明
name	是	String	连接名称，只能包含四种字符：英文字母、数字、中划线和下划线，且长度小于等于100个字符。连接名称不能重复。
type	是	String	连接类型，包含： <ul style="list-style-type: none"> <li>• DWS</li> <li>• DLI</li> <li>• SparkSQL</li> <li>• HIVE</li> <li>• RDS</li> <li>• CloudTable</li> <li>• HOST</li> </ul>
config	否	Map<String,String>	连接的配置项，不同类型的连接配置项不同。DLI类型的连接不需要配置。config参数，其他类型的连接请参考连接配置项说明。
description	否	String	描述，长度不能超过255个字符。

## 请求示例

查询连接列表。

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "total":1,
  "connections":[
    {
      "name":"connection1",
      "type":"DWS",
      "config":{
        "clusterName":"test",
        "userName":"dbadmin",
        "password":"*****",
        "kmsKey":"cdm-dlf",
        "agentName":"cdm-donotdelete",
        "sslEnable":false
      }
    }
  ]
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code":"DLF.3051",
  "error_msg":"The request parameter is invalid."
}
```

### 5.4.3 查询连接详情（待下线）

#### 📖 说明

连接管理能力由管理中心统一提供，相关接口已不再维护，推荐使用管理中心API进行连接管理。

## 功能介绍

查询指定连接的配置详情。

## 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

## URI

- URI格式  
GET /v1/{project\_id}/connections/{connection\_name}
- 参数说明

表 5-211 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
connection_name	是	String	连接名称

## 请求参数

表 5-212 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

## 响应参数

表 5-213 Connection 参数说明

参数名	是否必选	参数类型	说明
name	是	String	连接名称，只能包含四种字符：英文字母、数字、中划线和下划线，且长度小于等于100个字符。连接名称不能重复。
type	是	String	连接类型，包含： <ul style="list-style-type: none"> <li>DWS</li> <li>DLI</li> <li>SparkSQL</li> <li>HIVE</li> <li>RDS</li> <li>CloudTable</li> <li>HOST</li> </ul>
config	否	Map<String,String>	连接的配置项，不同类型的连接配置项不同。DLI类型的连接不需要配置。config参数，其他类型的连接请参考连接配置项说明。

参数名	是否必选	参数类型	说明
description	否	String	描述，长度不能超过255个字符。

## 请求示例

查询连接详情。

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/connection1
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "name": "connection1",
  "type": "DWS",
  "config": {
    "clusterName": "test",
    "userName": "dbadmin",
    "password": "*****",
    "kmsKey": "cdm-dlf",
    "agentName": "cdm-donotdelete",
    "sslEnable": false
  }
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.6322",
  "error_msg": "The data connection does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.4.4 修改连接（待下线）

### 说明

连接管理能力由管理中心统一提供，相关接口已不再维护，推荐使用管理中心API进行连接管理。

## 功能介绍

编辑修改连接的配置。

## 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

## URI

- URI格式  
PUT /v1/{project\_id}/connections/{connection\_name}?ischeck=true
- 参数说明

表 5-214 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
connection_name	是	String	连接名称。
ischeck	否	String	是否校验，默认否。

## 请求参数

表 5-215 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

表 5-216 Connection 参数说明

参数名	是否必选	参数类型	说明
name	是	String	连接名称，只能包含五种字符：英文字母、数字、中划线和下划线，且长度小于等于100个字符。连接名称不能重复。

参数名	是否必选	参数类型	说明
type	是	String	连接类型，包含： <ul style="list-style-type: none"> <li>• DWS</li> <li>• DLI</li> <li>• SparkSQL</li> <li>• HIVE</li> <li>• RDS</li> <li>• CloudTable</li> <li>• HOST</li> </ul>
config	否	Map<String,String>	连接的配置项，不同类型的连接配置项不同。DLI类型的连接不需要配置。config参数，其他类型的连接请参考连接配置项说明。
description	否	String	描述，长度不能超过255个字符。

## 响应参数

无

## 请求示例

修改连接。

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/connection1?ischeck=true
{
  "name":"connection1",
  "type":"DWS",
  "config":{
    "clusterName":"test",
    "userName":"dbadmin",
    "password":"*****",
    "kmsKey":"cdm-dlf",
    "agentName":"cdm-donotdelete",
    "sslEnable":false
  }
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code":"DLF.6322",
  "error_msg":"The data connection does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.4.5 删除连接（待下线）

### 说明

连接管理能力由管理中心统一提供，相关接口已不再维护，推荐使用管理中心API进行连接管理。

## 功能介绍

删除一个连接。

## 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

## URI

- URI格式  
DELETE /v1/{project\_id}/connections/{connection\_name}
- 参数说明

表 5-217 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
connection_name	是	String	连接名称。

## 请求参数

表 5-218 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>



## 响应参数

无

## 请求示例

删除一个连接。

```
DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/connection1
```

## 响应示例

成功响应

HTTP状态码 204

## 5.4.6 导出连接（待下线）

### 📖 说明

连接管理能力由管理中心统一提供，相关接口已不再维护，推荐使用管理中心API进行连接管理。

## 功能介绍

导出zip压缩包格式的连接文件，包括所有连接的定义。

## 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

## URI

- URI格式  
POST /v1/{project\_id}/connections/export
- 参数说明

表 5-219 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 5-220 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。

## 响应参数

响应消息的Content-Type是application/octet-stream，需要解析转化为一个文件，可以参考[解析响应消息体中的流样例代码](#)。响应是一个压缩文件，文件名称格式是DLF\_All\_DataConnections.zip，压缩文件里面的文件目录如下：

```
connections
├─{dwsConnection}.conn
```

## 请求示例

导出连接。

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/export
```

## 响应示例

- 成功响应

HTTP状态码 200

响应是一个压缩文件，压缩文件里面的文件目录如下：

```
connections
├─{dwsConnection}.conn
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.6322",
  "error_msg": "The data connection does not exist."
}
```

## 状态码

状态码请参见[状态码](#)。

## 5.4.7 导入连接（待下线）

### 📖 说明

连接管理能力由管理中心统一提供，相关接口已不再维护，推荐使用管理中心API进行连接管理。

## 功能介绍

将存放在OBS桶的连接文件导入到数据开发模块中，一次可以导入一个或多个连接。  
执行本接口之前，需要将连接文件放到OBS的桶中。

## 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

## URI

- URI格式  
POST /v1/{project\_id}/connections/import
- 参数说明

表 5-221 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

参数说明：

参数名	是否必选	参数类型	说明
path	是	String	有OBS场景：连接定义文件在OBS上的路径，连接定义文件的格式参考导出连接的响应消息。 无OBS场景：连接定义文件在本地的路径。
params	否	List<Params>	连接参数，默认值为空。
sameNamePolicy	否	String	重名策略 <ul style="list-style-type: none"> <li>• SKIP：跳过</li> <li>• OVERWRITE：覆盖。</li> </ul> 默认值是SKIP

Params 连接参数说明

参数名	是否必选	参数类型	说明
name	是	String	连接名称。
type	是	String	连接类型。

参数名	是否必选	参数类型	说明
params	否	Map<String,String>	连接参数，参数名称请参考每种类型的连接配置项说明，默认值为空。

## 响应参数

参数名	是否必选	参数类型	说明
taskId	是	String	任务ID，请调用查询系统任务接口获取导入状态。

## 请求示例

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/import
{
  "path": "obs://00zyx/2019-07-02/DLF_All_DataConnections.zip",
  "sameNamePolicy": "OVERWRITE",
  "params": [
    {
      "name": "DWS",
      "type": "DWS",
      "params": {
        "clusterName": "cluster1"
      }
    },
    {
      "name": "hive",
      "type": "HIVE",
      "params": {
        "clusterName": "mrs_ymcc",
        "connectionMethod": "agent",
        "userName": "admin",
        "agentName": "cdm-donotdelete",
        "kmsKey": "KMS-42ab"
      }
    }
  ]
}
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "taskId": "008aae2e675933c7016759418e870000"
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.0815",
  "error_msg": "Fail to read OBS file."
}
```

## 状态码

状态码请参见[状态码](#)。

# 6 数据开发 API ( V2 )

## 6.1 作业开发 API

### 6.1.1 创建补数据实例

#### 功能介绍

创建一个补数据实例。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
POST /v2/{project\_id}/factory/supplement-data
- 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 6-1 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	是	String	IAM Token 最小长度：0 最大长度：4096
Content-Type	是	String	发送的实体的MIME类型。

表 6-2 Body 参数说明

参数名	是否必选	参数类型	说明
name	是	String	补数据任务的名称。
job_name	是	String	作业名称，即补数据任务需要补哪个作业的名称。
start_date	是	String	补数据的开始日期，格式如"2023-08-21T00:00:00 +08"
end_date	是	String	补数据的结束日期，格式如"2023-08-21T23:59:59 +08"
parallel	是	int	补数据实例的并行周期数，取值范围[1,5]。设置同时执行的补数据实例数量，最多可同时执行5个实例。
depend_jobs	否	List<JobDTO>	参考表6-3。
is_day_granularity	否	Boolean	是否按照天粒度补数据，true表示是，false表示否。
priority	否	int	设置补数据作业的优先级。当系统资源不充足时，可以优先满足优先级较高的作业的计算资源，数字越大优先级越高，当前只支持对DLI SQL算子设置优先级。
is_stop_when_fail	否	Boolean	补数据作业失败时是否停止，true表示停止，false表示不停止。

参数名	是否必选	参数类型	说明
reverse_order	否	int	是否按照日期倒序补数据，取值范围[0,1]，0表示正序补数据，1表示倒序补数据。
force	否	String	是否强制补数据，true表示强制补数据，同一个作业出现在2个补数据实例时，也可以进行补数据；false表示不能强制补数据。
supplement_data_run_time	否	Object	补数据时间段，当前仅支持每天，如果没有补数据时间，则默认为"00:00-00:00"，参考表6-4。
supplement_data_instance_time	否	Object	支持补离散时间任务，参考表6-5。

表 6-3 JobDTO 参数说明

参数名	是否必选	参数类型	说明
job_name	否	String	作业名称。
workspace_id	否	String	工作空间ID。

表 6-4 SupplementDataRunTime 参数说明

参数名	是否必选	参数类型	说明
time_of_day	否	String	每天的可补数据时间段，如：每天的10点15分到23点30分，表示：10:15-23:30
day_of_week	否	String	每周的星期几可以补数据，如：每周一，周二的每天10点15分到23点30分
day_of_month	否	String	每个月的哪几天可以补数据，如每月1号，3号，表示：1,3

表 6-5 SupplementDataInstanceTime 参数说明

参数名	是否必选	参数类型	说明
days	否	array[string]	支持离散的天。
time_of_day	否	String	指定天中的时间段。



## 响应参数

无

## 请求示例

创建一个名称为P\_job\_6796\_20230823\_9991的补数据实例，依赖的作业名为job\_5799。

```
POST /v2/62099355b894428e8916573ae635f1f9/factory/supplement-data
```

```
{
  "name": "P_job_6796_20230823_9991",
  "job_name": "job_5799",
  "start_date": "2023-08-23T00:00:00 +08",
  "end_date": "2023-08-23T23:59:59 +08",
  "parallel": 1,
  "depend_jobs": [],
  "is_day_granularity": false,
  "priority": 0,
  "is_stop_when_fail": false,
  "reverse_order": 0,
  "supplement_data_run_time": {
    "time_of_day": "01:03-02:00"
  },
  "supplement_data_instance_time": {
    "days": ["2023/09/05-2023/09/05"],
    "time_of_day": "00:00-23:59"
  },
  "force": "true"
}
```

## 响应示例

- 成功响应  
HTTP状态码 200
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.3051",
  "error_msg": "The request parameter is invalid."
}
```

## 6.1.2 查询补数据实例

### 功能介绍

查询补数据实例列表，支持分页查询。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式

GET /v2/{project\_id}/factory/supplement-data?  
sort={sort}&page={page}&size={size}&name={name}&user\_name={user\_name}  
&status={status}&start\_date={start\_date}&end\_date={end\_date}

• 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
name	否	String	补数据实例名称
user_name	否	String	用户名
status	否	String	实例状态： <ul style="list-style-type: none"> <li>• SUCCESS：成功</li> <li>• RUNNING：运行中</li> <li>• CANCEL：取消</li> </ul>
sort	否	String	排序字段： <ul style="list-style-type: none"> <li>• desc：创建时间按照降序展示</li> <li>• asc：创建时间按照升序展示</li> </ul> 默认值：desc
page	否	Integer	分页列表的起始页，默认值为0。取值范围大于等于0。
size	否	Integer	分页返回结果，指定每页最大记录数。默认值：10。
start_date	否	Long	查询作业的开始日期，13位时间戳。
end_date	否	Long	查询作业的结束日期，13位时间戳。

## 请求参数

表 6-6 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	是	String	IAM Token 最小长度：0 最大长度：4096

## 响应参数

表 6-7 参数说明

参数名	是否必选	参数类型	说明
total	是	Integer	作业数量。
success	是	Boolean	true或者false。
msg	是	String	success。
rows	是	List<row>	包含若干补数据实例信息，参考表6-8。

表 6-8 row 数据结构说明

参数名	是否必选	参数类型	说明
job_list	是	List<String>	补数据作业名称，可能有依赖的作业，故会有多个作业的情况。
name	是	String	补数据实例名称。
user_name	是	String	用户名称。
type	是	int	触发补数据的类型，取值范围[0, 1]。0代表作业监控界面触发的补数据，1代表恢复动作触发的补数据。
start_date	是	Long	作业的开始日期，13位时间戳。
end_date	是	Long	作业的结束日期，13位时间戳。
parallel	是	int	补数据实例的并行周期数，取值范围[1,5]。
status	是	String	实例状态： <ul style="list-style-type: none"> <li>• SUCCESS: 成功</li> <li>• RUNNING: 运行中</li> <li>• CANCEL: 取消</li> </ul>
submitted_date	是	Long	作业创建时间。
supplement_data_run_time	否	supplement_data_run_time object	补数据时间段，当前仅支持每天，如果没有补数据时间，则默认为"00:00-00:00"，参考表6-9。

参数名	是否必选	参数类型	说明
supplement_data_instance_time	否	supplement_data_instance_time object	支持补离散时间任务，参考表6-10。

表 6-9 SupplementDataRunTime 参数说明

参数名	是否必选	参数类型	说明
time_of_day	是	String	每天的可补数据时间段，如：每天的10点15分到23点30分，表示：10:15-23:30。
day_of_week	否	String	每周的星期几可以补数据，如：每周一，周三的每天10点15分到23点30分。
day_of_month	否	String	每个月的哪几天可以补数据，如每月1号，3号，表示：1,3。

表 6-10 SupplementDataInstanceTime 参数说明

参数名	是否必选	参数类型	说明
days	否	array[string]	支持离散的天。
time_of_day	否	String	指定天中的时间段。

## 请求示例

查询作业列表。

```
GET /v2/62099355b894428e8916573ae635f1f9/factory/supplement-data
```

## 响应示例

- 成功响应 HTTP状态码200

```
{
  "msg": "success",
  "rows": [
    {
      "end_date": 1692633599000,
      "job_list": [
        "job_8810",
        "job_1000"
      ]
    }
  ],
}
```

```

"name": "P_job_8810_20230821_175711",
"parallel": 1,
"start_date": 1692547200000,
"status": "RUNNING",
"submitted_date": 1692611566436,
"supplement_data_run_time": {
  "time_of_day": "00:00-00:00"
},
"supplement_data_instance_time": {},
"type": 0,
"user_name": "user_test"
}
],
"success": true,
"total": 1
}

```

- 失败响应

HTTP状态码 400

```

{
  "error_code": "DLF.3051",
  "error_msg": "The request parameter is invalid."
}

```

## 6.1.3 停止补数据实例

### 功能介绍

停止指定的补数据实例。只有补数据实例的状态是运行中，才能停止补数据实例。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v2/{project\_id}/factory/supplement-data/{instance\_name}/stop
- 参数说明

表 6-11 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_name	是	String	补数据实例名称。

## 请求参数

表 6-12 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 • 如果不设置该参数，默认查询default工作空间下的数据。 • 如果需要查询其他工作空间的数据，需要带上该消息头。
X-Auth-Token	是	String	IAM Token 最小长度：0 最大长度：4096

## 响应参数

无

## 请求示例

```
POST /v2/62099355b894428e8916573ae635f1f9/factory/supplement-data/P_job_8002_203_aa/stop
```

## 响应示例

- 成功响应  
HTTP状态码 200
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF0113",
  "error_msg": "The supplement [P_job_1000_20230821_1540182] does not exist."
}
```

### 6.1.4 修改作业名称

#### 功能介绍

对作业名称进行重命名操作。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
PUT /v2/{project\_id}/factory/jobs/{job\_name}/rename

- 参数说明

表 6-13 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	原作业名称。

## 请求参数

表 6-14 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	否	String	使用Token认证时该字段必选，用户Token的长度要求如下： 最小长度： <b>0</b> 最大长度： <b>4096</b>
Content-Type	否	String	有Body体的情况下必选，无Body体的情况下则无需填写和校验。 示例：Content-Type: application/json;charset=UTF-8
Authorization	否	String	使用AK/SK进行认证时该字段必选。
Host	否	String	使用AK/SK进行认证时该字段必选。

表 6-15 Body 参数说明

参数名	是否必选	参数类型	说明
name	是	String	新的作业名称。

## 响应参数

表 6-16 参数说明

参数名	是否必选	参数类型	说明
is_success	是	Boolean	true或者false
status_code	是	Integer	状态响应码

## 请求示例

```
PUT /v2/62099355b894428e8916573ae635f1f9/factory/jobs/job_2284221/rename
{
  "name": "job_4269F"
}
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "is_success": true,
  "status_code": 200
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.30100",
  "error_msg": "Job[job_2284221] does not exist"
}
```

## 6.1.5 查询发布包列表

### 功能介绍

用于查询发布包的列表信息。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v2/{project\_id}/factory/release-packages
- 参数说明



表 6-17 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 6-18 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	否	String	使用Token认证时该字段必选，用户Token的长度要求如下： 最小长度： <b>0</b> 最大长度： <b>4096</b>
Content-Type	否	String	有Body体的情况下必选，无Body体的情况下则无需填写和校验。 示例：Content-Type: application/json;charset=UTF-8
Authorization	否	String	使用AK/SK进行认证时该字段必选。
Host	否	String	使用AK/SK进行认证时该字段必选。

表 6-19 Body 参数说明

参数名	是否必选	参数类型	说明
key_word	否	String	包名package_name关键字。
apply_user_name	否	String	申请人名称。
deploy_user_name	否	String	发布人名称。
apply_begin_time	否	Long	申请开始时间，13位时间戳。

参数名	是否必选	参数类型	说明
apply_end_time	否	Long	申请结束时间, 13位时间戳。
deploy_begin_time	否	Long	发布开始时间, 13位时间戳。
deploy_end_time	否	Long	发布结束时间, 13位时间戳。
apply_user_name_filter	否	array[string]	申请人名称集合, 根据该字段筛选, 如果选择了apply_user_name, 则该名称必须包含在集合内。
deploy_user_name_filter	否	array[string]	发布人名称集合, 根据该字段筛选, 如果选择了deploy_user_name, 则该名称必须包含在集合内。
deploy_statuses_filter	否	array[integers]	发布状态集合。 1: 待审批 2: 成功 3: 失败 5: 发布中
sorted_direction	否	String	排序方向, 默认是desc。
order_column	否	String	排序字段, 默认是apply_timestamp。
limit	否	Integer	分页返回结果, 默认是10。
offset	否	Integer	分页的起始页, 默认值为0, 取值范围大于等于0。

## 响应参数

表 6-20 参数说明

参数名	是否必选	参数类型	说明
data	是	array[object]	发布包信息, 参考 <a href="#">表6-21</a> 。
total	是	Integer	发布包个数。

表 6-21 发布包信息的参数说明

参数名	是否必选	参数类型	说明
apply_timestamp	否	Long	申请时间，13位时间戳。
apply_user_id	否	String	申请人id。
apply_user_name	否	String	申请人名称。
delete_flag	否	integer	是否删除。 0: 没有删除 1: 删除
deploy_status	否	integer	发布状态。 1: 待审批 2: 成功 3: 失败 5: 发布中
deploy_timestamp	否	Long	发布时间，13位时间戳。
deploy_user_id	否	String	发布人id。
deploy_user_name	否	String	发布人名称。
package_approvers	否	array[object]	发布包审批信息，参考 <a href="#">表6-22</a> 。
package_id	否	String	发布包id。
package_name	否	String	发布包名称。
project_id	否	String	项目ID和空间信息，以 项目ID-workspace-空间ID 拼接。
workspace_id	否	String	发布包所在空间ID。

表 6-22 发布包审批信息的参数说明

参数名	是否必选	参数类型	说明
user_id	否	String	发布包审批人id。
user_name	否	String	发布包审批人名称。

## 请求示例

```
POST /v2/62099355b894428e8916573ae635f1f9/factory/packages
{
  "key_word": "",
  "apply_user_name": "",
  "deploy_user_name": "",
  "apply_begin_time": 1699113600999,
  "apply_end_time": 1700323199999,
  "deploy_begin_time": "",
  "deploy_end_time": "",
  "apply_user_name_filter": "",
  "deploy_user_name_filter": "",
  "deploy_status_filter": "",
  "sorted_direction": "desc",
  "order_column": "apply_timestamp",
  "limit": 10,
  "offset": 0
}
```

## 响应示例

- 成功响应  
HTTP状态码 200

```
{
  "data": [
    {
      "apply_timestamp": 1700288912934,
      "apply_user_id": "7b71e498e75d44048c9a22dd3c54f978",
      "apply_user_name": "ei_dlf_l00341563",
      "delete_flag": 0,
      "deploy_status": 3,
      "deploy_timestamp": 1700289155129,
      "deploy_user_id": "7b71e498e75d44048c9a22dd3c54f978",
      "deploy_user_name": "ei_dlf_l00341563",
      "package_approvers": [
        {
          "user_id": "7b71e498e75d44048c9a22dd3c54f978",
          "user_name": "ei_dlf_l00341563"
        }
      ],
      "package_id": "4",
      "package_name":
      "ads_supplier_inventory_info__ads_supplier_inventory_info_20231118143537",
      "project_id": "62099355b894428e8916573ae635f1f9-
      workspace-45be704b28cb4f25a39221169fd3e27e",
      "workspace_id": "45be704b28cb4f25a39221169fd3e27e"
    },
    {
      "apply_timestamp": 1700288777664,
      "apply_user_id": "7b71e498e75d44048c9a22dd3c54f978",
      "apply_user_name": "ei_dlf_l00341563",
      "delete_flag": 0,
      "deploy_status": 2,
      "deploy_timestamp": 1700289161361,
      "deploy_user_id": "7b71e498e75d44048c9a22dd3c54f978",
      "deploy_user_name": "ei_dlf_l00341563",
      "package_approvers": [
        {
          "user_id": "7b71e498e75d44048c9a22dd3c54f978",
          "user_name": "ei_dlf_l00341563"
        }
      ],
      "package_id": "3",
      "package_name": "aa_20231118143318",
      "project_id": "62099355b894428e8916573ae635f1f9-
      workspace-45be704b28cb4f25a39221169fd3e27e",
      "workspace_id": "45be704b28cb4f25a39221169fd3e27e"
    }
  ]
}
```

```
    }
  ],
  "total": 2
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.3051",
  "error_msg": "The request parameter is invalid. "
}
```

## 6.1.6 查询指定发布包详情

### 功能介绍

用于查询指定的发布包详情信息。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
GET /v2/{project\_id}/factory/release-packages/{package\_id}
- 参数说明

表 6-23 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
package_id	是	String	发布包ID。获取方法请参考“ <a href="#">查询发布包列表</a> ”的响应参数“package_id”。

### 请求参数

表 6-24 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>

参数名	是否必选	参数类型	说明
X-Auth-Token	否	String	使用Token认证时该字段必选，用户Token的长度要求如下： 最小长度： <b>0</b> 最大长度： <b>4096</b>
Content-Type	否	String	有Body体的情况下必选，无Body体的情况下则无需填写和校验。 示例：Content-Type: application/json;charset=UTF-8
Authorization	否	String	使用AK/SK进行认证时该字段必选。
Host	否	String	使用AK/SK进行认证时该字段必选。

## 响应参数

表 6-25 参数说明

参数名	是否必选	参数类型	说明
release_package	是	object	发布包详情，参考 <a href="#">表6-26</a> 。
task_details	是	array[object]	发布任务详情，参考 <a href="#">表6-27</a> 。

表 6-26 发布包详情的参数说明

参数名	是否必选	参数类型	说明
apply_timestamp	否	Long	申请时间，13位时间戳。
apply_user_id	否	String	申请人id。
apply_user_name	否	String	申请人名称。
delete_flag	否	integer	是否删除。 0：没有删除 1：删除

参数名	是否必选	参数类型	说明
deploy_status	否	integer	发布状态。 1: 待审批 2: 成功 3: 失败 5: 发布中
deploy_timestamp	否	Long	发布时间, 13位时间戳。
deploy_user_id	否	String	发布人id。
deploy_user_name	否	String	发布人名称。
package_approvers	否	array[object]	发布包审批信息, 参考表6-28。
package_id	否	String	发布包id。
package_name	否	String	发布包名称。
project_id	否	String	项目ID和空间信息, 以项目ID-workspace-空间ID拼接。
workspace_id	否	String	发布包所在空间ID。

表 6-27 发布任务详情的参数说明

参数名	是否必选	参数类型	说明
deploy_status	否	integer	发布状态。 1: 待审批 2: 成功 3: 失败 5: 发布中
deployed_version	否	integer	已发布节点版本。
item_name	否	String	发布任务名称。
pending_item_id	否	String	发布任务ID。
pending_version	否	integer	当前节点版本。
script_id	否	String	脚本ID。

参数名	是否必选	参数类型	说明
task_id	否	String	作业ID。
start_job_statuses	否	integer	作业启动状态。 2: 成功 3: 失败
submit_timestamp	否	Long	提交时间戳, 13位时间戳。
submit_user_id	否	String	提交人id。
submit_user_name	否	String	提交人名称。
task_type	否	integer	任务类型。 1: 作业 2: 脚本 3: 资源
update_type	否	integer	变更类型, 默认值1。 1: 新增 2: 修改 3: 删除

表 6-28 发布包审批信息的参数说明

参数名	是否必选	参数类型	说明
user_id	否	String	发布包审批人id。
user_name	否	String	发布包审批人名称。

## 请求示例

GET /v2/62099355b894428e8916573ae635f1f9/factory/release-packages/1

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "release_package": {
    "apply_timestamp": 1700288912934,
    "apply_user_id": "7b71e498e75d44048c9a22dd3c54f978",
    "apply_user_name": "ei_dlf_l00341563",
    "delete_flag": 0,
    "deploy_status": 3,
```



```

"deploy_timestamp": 1700289155129,
"deploy_user_id": "7b71e498e75d44048c9a22dd3c54f978",
"deploy_user_name": "ei_dlf_l00341563",
"package_approvers": [
  {
    "user_id": "7b71e498e75d44048c9a22dd3c54f978",
    "user_name": "ei_dlf_l00341563"
  }
],
"package_id": "4",
"package_name": "ads_supplier_inventory_info_ads_supplier_inventory_info_20231118143537",
"project_id": "62099355b894428e8916573ae635f1f9-workspace-45be704b28cb4f25a39221169fd3e27e",
"workspace_id": "45be704b28cb4f25a39221169fd3e27e"
},
"task_details": [
  {
    "deploy_status": 3,
    "deployed_version": 0,
    "item_name": "ads_supplier_inventory_info_ads_supplier_inventory_info",
    "pending_item_id": "19",
    "pending_version": 1,
    "script_id": "ff808081899c04fa01899fc39cf61baa",
    "submit_timestamp": 1690602544380,
    "submit_user_id": "7b71e498e75d44048c9a22dd3c54f978",
    "submit_user_name": "ei_dlf_l00341563",
    "task_type": 2,
    "update_type": 1
  }
]
}

```

- 失败响应

HTTP状态码 400

```

{
  "error_code": "DLE.3051",
  "error_msg": "The request parameter is invalid. "
}

```

## 6.1.7 设置作业标签

### 功能介绍

此接口可以设置作业标签。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v2/{project\_id}/factory/jobs/{job\_name}/tags
- 参数说明

表 6-29 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。

## 请求参数

表 6-30 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	是	String	IAM Token 最小长度： <b>0</b> 最大长度： <b>4096</b>
Content-Type	否	String	有Body体的情况下必选，无Body体的情况下则无需填写和校验。 示例：Content-Type: application/json;charset=UTF-8

表 6-31 Body 参数说明

参数名	是否必选	参数类型	说明
tag_names	否	List<String>	标签列表。若为空表示清空作业标签。 样例值：["tag1", "tag2"]

## 响应参数

无

## 请求示例

为作业myJob设置标签“tag1”。

```
POST /v2/b384b9e9ab9b4ee8994c8633aabc9505/factory/jobs/myJob/tags
{
```

```
"tag_names": ["tag1"]
}
```

清空作业myJob的所有标签。

```
POST /v2/b384b9e9ab9b4ee8994c8633aabc9505/factory/jobs/myJob/tags
{
  "tag_names": []
}
```

## 响应示例

- 成功响应  
HTTP状态码 204
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.100",
  "error_msg": "The job does not exists."
}
```

## 6.1.8 查询告警通知记录

### 功能介绍

此接口可以查询指定时间段内的告警通知记录。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
GET /v2/{project\_id}/factory/alarm-info?  
start\_time={start\_time}&end\_time={end\_time}&limit={limit}&offset={offset}
- 参数说明

表 6-32 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
start_time	否	Long	告警的开始时间，默认当前时间的前一个小时，13位时间戳。
end_time	否	Long	告警的最后时间，默认为当前时间，13位时间戳，支持查询最近一周的时间段
offset	否	Integer	分页的起始页，默认值为0。取值范围大于等于0。

参数名	是否必选	参数类型	说明
limit	否	Integer	分页返回结果，指定每页最大记录数。默认值为100，取值范围为[1, 1000]

## 请求参数

表 6-33 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	否	String	IAM Token 最小长度：0 最大长度：4096

## 响应参数

表 6-34 参数说明

参数名	是否必选	参数类型	说明
alarm_info	否	List<AlarmInfo>	通知记录信息，请参考 <a href="#">表6-35</a> 。
total	否	Integer	通知记录数量。

表 6-35 AlarmInfo 数据结构

参数名	是否必选	参数类型	说明
alarm_time	否	Long	告警通知时间。
job_name	否	String	作业名称。

参数名	是否必选	参数类型	说明
schedule_type	否	Integer	作业实例调度方式。取值范围如下： 0: 正常调度 2: 手工调度 5: 补数据 6: 子作业调度 7: 单次调度
send_msg	否	String	发送信息。
plan_time	否	Long	计划时间。
remind_type	否	Integer	告警通知类型，取值范围如下： 0: 运行成功 1: 运行异常/失败 3: 未完成 4: 资源繁忙 12: 周期未完成 13: 运行取消 14: 失败作业重跑成功 15: 作业改动
send_status	否	Integer	发送状态，取值范围如下： 0: 发送成功 1: 发送失败
job_id	否	Long	作业ID。

## 请求示例

```
GET /v2/b384b9e9ab9b4ee8994c8633aabc9505/factory/alarm-info?
start_time=1702378620000&end_time=1702710169595&limit=3&offset=0
```

## 响应示例

- 成功响应  
HTTP状态码 200

```
{
  "alarm_info": [
    {
      "alarm_time": 1702710169460,
      "job_name": "job_0998",
      "schedule_type": 0,
      "send_msg": "success",
      "plan_time": 1702378440000,
      "remind_type": 0,
      "send_status": 0,
      "job_id": 911925
    }
  ],
  {
```

```
"job_name": "job_0998",
"schedule_type": 0,
"send_msg": "success",
"remind_type": 0,
"send_status": 0,
"alarm_time": 1702710169375,
"plan_time": 1702378260000,
"job_id": 911925
}
],
"total": 20
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF0810",
  "error_msg": "Task does not exist."
}
```

## 6.1.9 发布任务包

### 功能介绍

目前支持发布包管理中相关包的发布，支持发布多个任务包。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
POST /v2/{project\_id}/factory/release-packages/deploy
- 参数说明

表 6-36 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 6-37 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	否	String	IAM Token 最小长度： <b>0</b> 最大长度： <b>4096</b>
Content-Type	否	String	有Body体的情况下必选，无Body体的情况下则无需填写和校验。 示例：Content-Type: application/json;charset=UTF-8

表 6-38 请求 Body 参数

参数名	是否必选	参数类型	说明
package_ids	是	array[string]	发布包ID集合，不能为空，且一次不能超过100个。发布包ID请通过“查询发布包列表”接口获取。
startup_mode	否	Integer	发布后是否立即启动作业。取值范围为0和1，默认为1 1：发布成功后立即启动作业 0：不立即启动

## 响应参数

表 6-39 响应参数说明

参数名	是否必选	参数类型	说明
deploy_package_details	否	List<Object>	发布包信息，请参考 <a href="#">表6-40</a> 。

表 6-40 deploy\_package\_details 数据结构

参数名	是否必选	参数类型	说明
asy_subtask_num	否	Integer	总的异步执行的子任务个数。
asy_task_id	否	String	异步作业id, 返回给前台轮询结果。
package_id	否	Long	发布包ID。

## 请求示例

```
POST /v2/b384b9e9ab9b4ee8994c8633aabc9505/factory/release-packages/deploy
{
  "package_ids": ["23"],
  "startup_mode": 0
}
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "deploy_package_details": [
    {
      "asy_subtask_num": 1,
      "asy_task_id": "ff8080818c81c8ab018c8b43f85d0008",
      "package_id": 23
    }
  ]
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.0810",
  "error_msg": "Task does not exist."
}
```

## 6.1.10 撤销任务包

### 功能介绍

目前支持发布包管理中相关包的撤销, 支持撤销多个任务包。

### 调试

您可以在[API Explorer](#)中调试该接口, 支持自动认证鉴权。

### URI

- URI格式  
POST /v2/{project\_id}/factory/release-packages/unpack



- 参数说明

表 6-41 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 6-42 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	否	String	IAM Token 最小长度： <b>0</b> 最大长度： <b>4096</b>
Content-Type	否	String	有Body体的情况下必选，无Body体的情况下则无需填写和校验。 示例：Content-Type: application/json;charset=UTF-8

表 6-43 请求 Body 参数

参数名	是否必选	参数类型	说明
package_ids	否	array[string]	发布包ID集合。

## 响应参数

无

## 请求示例

```
POST /v2/b384b9e9ab9b4ee8994c8633aabc9505/factory/release-packages/unpack
{"package_ids":["12","13"]}
```

## 响应示例

- 成功响应  
HTTP状态码 200
- 失败响应  
HTTP状态码 400

```
{
  "error_code": "DLF.0810",
  "error_msg": "Task does not exist."
}
```

### 6.1.11 查询实例运行状态

#### 功能介绍

此接口可以查询实例运行状态，包括当天、昨天、前天以及近7天的实例状态情况。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

#### URI

- URI格式  
GET /v2/{project\_id}/factory/monitor/task-statistics-overview?  
is\_own={is\_own}&query\_days={query\_days}
- 参数说明

表 6-44 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
is_own	否	String	是否查询当前用户的实例，默认为false，表示查询全部用户实例，为true时，表示查询当前用户的实例。
query_days	否	String	查询的天数，取值范围为：today、yesterday、before_yesterday、all，默认为today，表示查询今天的数据，支持查询近7天的数据。 today：查询当天的实例状态数量， yesterday：查询昨天的实例状态数量， before_yesterday：查询前天的实例状态数量， all查询7天前到当天的实例状态总量。

## 请求参数

表 6-45 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	否	String	IAM Token 最小长度： <b>0</b> 最大长度： <b>4096</b>

## 响应参数

表 6-46 响应参数说明

参数名	是否必选	参数类型	说明
fail_count	否	Integer	失败的实例数量
force_success_count	否	Integer	强制成功的实例数量
freeze_count	否	Integer	冻结的实例数量
ignore_success_count	否	Integer	忽略失败的实例数量
manual_stop_count	否	Integer	取消的实例数量
pause_count	否	Integer	暂停的实例数量
running_count	否	Integer	运行中的实例数量
running_exception_count	否	Integer	异常的实例数量。
skip_count	否	Integer	跳过的实例数量
success_count	否	Integer	运行成功的实例数量
waiting_confirmation_count	否	Integer	待确认执行的实例数量
waiting_count	否	Integer	等待运行的实例数量
total_count	否	Integer	实例总数

## 请求示例

```
GET /v2/b384b9e9ab9b4ee8994c8633aabc9505/factory/monitor/task-statistics-overview?  
is_own=false&query_days=today
```

## 响应示例

- 成功响应

HTTP状态码 200

```
{  
  "fail_count": 0,  
  "force_success_count": 0,  
  "freeze_count": 0,  
  "ignore_success_count": 0,  
  "manual_stop_count": 0,  
  "pause_count": 0,  
  "running_count": 0,  
  "running_exception_count": 0,  
  "skip_count": 0,  
  "success_count": 1,  
  "total_count": 1,  
  "waiting_confirm_count": 0,  
  "waiting_count": 0  
}
```

- 失败响应

HTTP状态码 400

```
{  
  "error_code": "DLF.21113",  
  "error_msg": "The task query parameters [today2] are incorrect. Currently, the supported types of  
parameters are today, yesterday, before_yesterday, and all."  
}
```

## 6.1.12 查询任务完成情况

### 功能介绍

查询任务完成情况。包括今天，昨天以及近7天的平均任务完成情况。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
GET /v2/{project\_id}/factory/monitor/task-completion-data?type={type}
- 参数说明

表 6-47 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

参数名	是否必选	参数类型	说明
type	否	String	<p>查询任务的类型，默认为all，查询所有任务。类型如下：Dummy、CDM Job、MRS Hive SQL、MRS Spark SQL、DLI SQL、DLI Spark、Python、DWS SQL、Shell等作业算子。</p> <p><b>说明</b> 此处只是列举一些算子，以实际为准。 该接口只统计运行成功的实例，每小时统计一次，任务表示作业中的算子。</p>

## 请求参数

表 6-48 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	<p>工作空间id。</p> <ul style="list-style-type: none"> <li>如果不设置该参数，默认查询default工作空间下的数据。</li> <li>如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>
X-Auth-Token	否	String	<p>IAM Token</p> <p>最小长度：0</p> <p>最大长度：4096</p>

## 响应参数

表 6-49 响应参数说明

参数名	是否必选	参数类型	说明
yesterday	否	array[object]	昨天的任务信息，请参考 <a href="#">task_record结构体</a> 。
average	否	array[object]	近7天的平均任务信息，请参考 <a href="#">task_record结构体</a> 。
today	否	array[object]	今天的任务信息，请参考 <a href="#">task_record结构体</a> 。

表 6-50 task\_record 参数说明

参数名	是否必选	参数类型	说明
record_time	否	Long	整时的时间点，13位时间戳。
task_completion_num	否	Long	当前时间点完成的任务数量。

## 请求示例

GET /v2/b384b9e9ab9b4ee8994c8633aabc9505/factory/monitor/task-completion-data?type=all

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "yesterday": [
    {
      "record_time": 1704124800000,
      "task_completion_num": 0
    },
    {
      "record_time": 1704128400000,
      "task_completion_num": 0
    },
    {
      "record_time": 1704132000000,
      "task_completion_num": 0
    }
  ],
  "average": [
    {
      "record_time": 1704124800000,
      "task_completion_num": 0
    },
    {
      "record_time": 1704128400000,
      "task_completion_num": 0
    },
    {
      "record_time": 1704132000000,
      "task_completion_num": 0
    }
  ],
  "today": [
    {
      "record_time": 1704124800000,
      "task_completion_num": 0
    },
    {
      "record_time": 1704128400000,
      "task_completion_num": 0
    },
    {
      "record_time": 1704132000000,
      "task_completion_num": 0
    }
  ]
}
```

- 失败响应

HTTP状态码 400

```
{
  "error_code": "DLF.3051",
  "error_msg": "The request parameter is invalid. "
}
```

## 6.1.13 查询指定作业的实例列表

### 功能介绍

根据作业名称精确查询作业实例列表。

### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。

### URI

- URI格式  
GET /v2/{project\_id}/factory/jobs/{job\_name}/instances/detail  
?  
min\_plan\_time={min\_plan\_time}&max\_plan\_time={max\_plan\_time}&limit={limit}&offset={offset}&status={status}&instance\_type={instance\_type}&force\_success={force\_success}&ignore\_success={ignore\_success}
- 参数说明

表 6-51 URI 参数说明

参数名	是否必选	参数类型	说明
project_id	是	String	项目编号，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_name	是	String	作业名称。 <ul style="list-style-type: none"> <li>如果要查询指定批处理作业的实例列表，job_name就是批处理作业名称；</li> <li>如果要查询实时作业下某个节点关联的子作业，job_name格式为[实时作业名称]_[节点名称]。</li> </ul>
min_plan_time	否	Long	返回作业实例开始时间大于min_plan_time的作业实例，单位为毫秒ms，默认设置为查询当天0点，最大可支持查询一个月。
max_plan_time	否	Long	返回作业实例开始时间小于max_plan_time的作业实例，单位为毫秒ms，默认设置为当前时间。
limit	否	int	分页返回结果，指定每页最大记录数。 范围[1,1000] 默认值：10

参数名	是否必选	参数类型	说明
offset	否	int	分页的起始页，默认值为0。取值范围大于等于0。
status	否	String	实例运行状态： <ul style="list-style-type: none"> <li>• waiting：等待运行</li> <li>• running：运行中</li> <li>• success：运行成功</li> <li>• fail：运行失败</li> <li>• running-exception：运行异常</li> <li>• skip-by-depend：跳过</li> <li>• freeze：冻结</li> <li>• pause：暂停</li> <li>• manual-stop：取消</li> </ul> 默认查全部
instance_type	否	int	作业调度方式： <ul style="list-style-type: none"> <li>• 0：正常调度</li> <li>• 2：手工调度</li> <li>• 5：补数据</li> <li>• 6：子作业调度</li> <li>• 7：单次调度</li> </ul> 默认查全部
force_success	否	boolean	status为success的时候使用，true则筛选出强制成功的作业实例 默认值：false
ignore_success	否	boolean	status为success的时候使用，true则筛选出忽略失败的作业实例 默认值：false

## 请求参数

表 6-52 请求 Header 参数

参数名	是否必选	参数类型	说明
workspace	否	String	工作空间id。 <ul style="list-style-type: none"> <li>• 如果不设置该参数，默认查询default工作空间下的数据。</li> <li>• 如果需要查询其他工作空间的数据，需要带上该消息头。</li> </ul>



## 响应参数

表 6-53 参数说明

参数名	是否必选	参数类型	说明
total	是	Long	总记录数
instances	是	List<Instance>	作业实例状态，参考表6-54。

表 6-54 Instance 属性说明

参数名	是否必选	参数类型	说明
job_name	是	String	作业名称。如果要查询指定批处理作业的实例列表，job_name就是批处理作业名称；如果要查询实时作业下某个节点关联的子作业，job_name格式为[实时作业名称]_[节点名称]。
job_id	否	Long	作业id。
job_instance_name	是	String	作业实例运行时日志记录的实例名称，非作业定义的名称。
status	是	String	状态： <ul style="list-style-type: none"> <li>waiting：等待运行</li> <li>running：运行中</li> <li>success：运行成功</li> <li>fail：运行失败</li> <li>running-exception：运行异常</li> <li>skip-by-depend：跳过</li> <li>freeze：冻结</li> <li>pause：暂停</li> <li>manual-stop：取消</li> </ul>
plan_time	是	Long	作业实例计划执行时间
start_time	是	Long	作业实例实际执行开始时间
end_time	否	Long	作业实例实际执行结束时间
execute_time	否	Long	执行耗时，单位：毫秒
instance_id	是	Long	作业实例ID，用于查询作业实例详情。

参数名	是否必选	参数类型	说明
submit_time	是	Long	作业提交运行时间
instance_type	是	int	作业调度方式： <ul style="list-style-type: none"> <li>• 0: 正常调度</li> <li>• 2: 手工调度</li> <li>• 5: 补数据</li> <li>• 6: 子作业调度</li> <li>• 7: 单次调度</li> </ul>
force_success	否	boolean	作业实例状态筛选为强制成功 默认值: false
ignore_success	否	boolean	作业实例状态筛选为忽略失败 默认值: false
version	否	int	作业实例版本。

## 请求示例

查询批处理作业job\_batch的实例列表

GET /v2/b384b9e9ab9b4ee8994c8633aabc9505/factory/jobs/job\_batch/instances/detail

## 响应示例

- 成功响应

HTTP状态码 200

```
{
  "total": 2,
  "instances": [
    {
      "end_time": 1551671598000,
      "execute_time": 0.3,
      "force_success": false,
      "ignore_success": false,
      "instance_id": 34765,
      "instance_type": 2,
      "job_id": 608627,
      "job_name": "job_batch",
      "job_instance_name": "job_batch",
      "plan_time": 1551671580000,
      "start_time": 1551671580000,
      "status": "success",
      "submit_time": 1550910278706,
      "version": 1
    },
    {
      "end_time": 1551671538000,
      "execute_time": 0.3,
      "force_success": false,
      "ignore_success": false,
      "instance_id": 34764,
      "instance_type": 5,
      "job_id": 608627,
      "job_name": "job_batch",
      "job_instance_name": "job_batch",

```

```

    "plan_time": 1551671520000,
    "start_time": 1551671521000,
    "status": "success",
    "submit_time": 1550910278706,
    "version": 1
  }}
}

```

- 失败响应

HTTP状态码 400

```

{
  "error_code": "DLF.0100",
  "error_msg": "The job does not exists."
}

```

## 6.1.14 重跑作业实例

### 功能介绍

支持重跑作业实例以及上下游的作业实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/factory/jobs/{job\_name}/instances/retry

表 6-55 路径参数

参数	是否必选	参数类型	描述
job_name	是	String	当前作业名称
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 6-56 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值），使用Token认证时必选，最小长度：0，最大长度：4096。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
Content-Type	是	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。
X-Project-Id	否	String	project_id, 项目编号。获取方法请参见 <a href="#">项目ID和账号ID</a> , 多project场景采用AK/SK认证的接口请求, 则该字段必选。

表 6-57 请求 Body 参数

参数	是否必选	参数类型	描述
retry_location	是	String	重跑开始位置, 取值有 firstNode、errorNode和 specifiedNode。
node_name	否	String	节点名称。
retry_task_version	是	String	使用的作业参数, 取值有 original_version和 current_version。
ignore_obs_monitor	否	Boolean	是否忽略obs监听, 默认为 true。
task_retrys	否	Array of <a href="#">task_retrys</a> objects	作业实例重跑参数, 当重跑当前实例类型时, 需要指定该参数的重跑信息, 重跑当前作业及其上下游作业实例类型不需要指定该参数。
begin_time	否	Long	实例开始时间, 当重跑当前作业及其上下游作业实例类型时, 该参数有效。
end_time	否	Long	实例结束时间, 当重跑当前作业及其上下游作业实例类型时, 该参数有效。
jobs	否	Array of <a href="#">jobs</a> objects	作业实例重跑参数, 当重跑当前作业及其上下游作业实例类型时, 需要指定该参数的重跑信息, 重跑当前实例类型不需要指定该参数。

参数	是否必选	参数类型	描述
concurrent	否	Integer	并行度，当重跑当前作业及其上下游作业实例类型时，该参数有效，默认值为1，取值范围为1到128。

表 6-58 task\_retrys

参数	是否必选	参数类型	描述
job_id	是	Long	作业id，当前重跑实例的作业id。
plan_time	是	Long	实例计划时间，13位时间戳，可通过查询作业实例列表接口获取。
submit_time	是	Long	实例提交时间，13位时间戳，可通过查询作业实例列表接口获取。

表 6-59 jobs

参数	是否必选	参数类型	描述
job_name	是	String	重跑的作业名称。
workspace_id	是	String	工作空间id

## 响应参数

状态码： 400

表 6-60 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

- 重跑作业为test\_job\_name的当前实例。

```
POST /v2/62099355b894428e8916573ae635f1f9/factory/jobs/test_job_name/instances/retry
{
  "retry_location": "errorNode",
```

```
"retry_task_version" : "original_version",
"ignore_obs_monitor" : true,
"task_retrys" : [ {
  "job_id" : 917567,
  "plan_time" : 1732601580000,
  "submit_time" : 1732239754684
} ]
}
```

- 重跑作业为test\_job\_name的当前作业及其上下游作业实例。

POST /v2/62099355b894428e8916573ae635f1f9/factory/jobs/test\_job\_name/instances/retry-task

```
{
  "retry_location" : "errorNode",
  "retry_task_version" : "current_version",
  "ignore_obs_monitor" : true,
  "begin_time" : 1732608520000,
  "end_time" : 1732612120000,
  "jobs" : [ {
    "job_name" : "job_test_smn2",
    "workspace_id" : "62fbba734e634c5087656fc3ff7adf34"
  } ],
  "concurrent" : 2
}
```

## 响应示例

**状态码： 400**

作业重跑失败的响应信息。

```
{
  "error_code" : "DLF.1000",
  "error_msg" : "Failed to find the job."
}
```

## 状态码

状态码	描述
200	作业重跑成功，无响应信息
400	作业重跑失败的响应信息。

# 7 管理中心 API

## 7.1 数据连接管理

### 7.1.1 查询数据连接列表

#### 功能介绍

查询数据连接列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/data-connections

表 7-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 7-2 Query 参数

参数	是否必选	参数类型	描述
name	否	String	数据连接名称，可通过本接口的全量 <a href="#">数据连接列表</a> 返回结果获取。

参数	是否必选	参数类型	描述
type	否	String	数据连接类型，可通过本接口的全量 <a href="#">数据连接列表</a> 返回结果获取。
limit	否	String	每次查询的数据连接条数的限制量。
offset	否	String	每次查询的数据连接条数的分页页码。

## 请求参数

表 7-3 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 200

表 7-4 响应 Body 参数

参数	参数类型	描述
count	Integer	当前分页返回数据连接记录数。
max_records	Integer	返回记录总数，一个工作空间最多只能创建的数据连接条数。
data_connection_lists	Array of <a href="#">ApigDataSourceView</a> objects	返回数据连接列表。



表 7-5 ApigDataSourceView

参数	参数类型	描述
dw_name	String	数据连接名称。
dw_type	String	数据连接类型。
create_user	String	数据连接创建者。
create_time	Number	数据连接创建时间，时间戳。
agent_id	String	CDM集群ID，获取方法请参见 <a href="#">查询集群列表</a> 。
agent_name	String	代理集群名称，获取方法请参见 <a href="#">查询集群列表</a> 。
dw_id	String	数据连接id。
qualified_name	String	数据连接限定名称。
description	String	数据连接描述。

状态码： 400

表 7-6 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-7 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

状态码： 400

```
{
  "error_code" : "DAYU.4402",
```

```
"error_msg" : "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDataconnectionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDataconnectionsRequest request = new ListDataconnectionsRequest();
        try {
            ListDataconnectionsResponse response = client.listDataconnections(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListDataconnectionsRequest()
        response = client.list_dataconnections(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.ListDataconnectionsRequest{}  
response, err := client.ListDataconnections(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回数据连接列表。
400	BadRequest。
500	Internal Server Error。

## 7.1.2 创建数据连接

### 功能介绍

创建数据连接。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/data-connections

表 7-8 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 7-9 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 7-10 请求 Body 参数

参数	是否必选	参数类型	描述
data_source_vos	否	Array of <a href="#">ApigDataSourceVo</a> objects	数据源结构体。

表 7-11 ApigDataSourceVo

参数	是否必选	参数类型	描述
dw_name	是	String	数据连接名称，创建时自定义，编辑时通过 <a href="#">数据连接列表</a> 获取。
dw_type	是	String	数据连接类型，如：DWS、DLI、HIVE、RDS、SparkSQL等，可在console界面查看所有数据连接类型，编辑时通过 <a href="#">数据连接列表</a> 获取。
dw_config	是	Object	连接动态变化配置项，每种连接略有区别，建议在界面进行调试。
agent_id	否	String	CDM集群ID，获取方法请参见 <a href="#">查询集群列表</a> 。
agent_name	否	String	代理集群名称，获取方法请参见 <a href="#">查询集群列表</a> 。
env_type	否	Integer	0：开发模式 1：生产模式，默认为0。
qualified_name	否	String	数据连接限定名称。

参数	是否必选	参数类型	描述
dw_id	否	String	数据连接id, 可通过 <a href="#">数据连接列表</a> 中获取。
create_user	否	String	数据连接创建者。
create_time	否	Number	数据连接创建时间, 时间戳。

## 响应参数

状态码： 400

表 7-12 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-13 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{
  "data_source_vos": [{
    "dw_config": {
      "clusterId": "353ff458-a560-413e-9b84-33f930cb8057",
      "clusterName": "mrs_3x_autotest_do_not_del",
      "dbUserName": "xxxxxx",
      "dbPassword": "xxxxxx",
      "kmsId": "a721616c-9a12-47b1-a805-3cfd3e63cd7",
      "kmsName": "KMS-1111",
      "mrsConnectionType": "agent"
    },
    "env_type": 0,
    "dw_type": "HIVE",
    "dw_name": "test_hive_01",
    "agent_id": "91f81a12-75c5-43ce-aab8-7149ecef3b17",
    "agent_name": "cdm-4autotest-nodelete"
  }]
}
```

## 响应示例

### 状态码： 200

Success, 数据连接创建成功。

```
{
  "data_connection_id" : "1c868514f45447a59c41485b70ae5e05"
}
```

### 状态码： 400

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

### 状态码： 500

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateConnectionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))
            .build();
        CreateConnectionsRequest request = new CreateConnectionsRequest();
        ApigDataSourcesVo body = new ApigDataSourcesVo();
        request.withBody(body);
        try {
```

```
        CreateConnectionsResponse response = client.createConnections(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectionsRequest()
        request.body = ApigDataSourcesVo(
        )
        response = client.create_connections(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```



```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateConnectionsRequest{
    request.Body = &model.ApigDataSourcesVo{
    }
}
response, err := client.CreateConnections(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，数据连接创建成功。
400	BadRequest。
500	Internal Server Error。

### 7.1.3 测试创建数据连接

#### 功能介绍

测试创建数据连接。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/{project\_id}/data-connections/validation

表 7-14 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 7-15 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 7-16 请求 Body 参数

参数	是否必选	参数类型	描述
dw_name	是	String	数据连接名称，创建时自定义，编辑时通过 <a href="#">数据连接列表</a> 获取。
dw_type	是	String	数据连接类型，如：DWS、DLI、HIVE、RDS、SparkSQL等，可在console界面查看所有数据连接类型，编辑时通过 <a href="#">数据连接列表</a> 获取。
dw_config	是	Object	连接动态变化配置项，每种连接略有区别，建议在界面进行调试。
agent_id	否	String	CDM集群ID，获取方法请参见 <a href="#">查询集群列表</a> 。
agent_name	否	String	代理集群名称，获取方法请参见 <a href="#">查询集群列表</a> 。
env_type	否	Integer	0：开发模式 1：生产模式，默认为0。
qualified_name	否	String	数据连接限定名称。

参数	是否必选	参数类型	描述
dw_id	否	String	数据连接id, 可通过 <a href="#">数据连接列表</a> 中获取。
create_user	否	String	数据连接创建者。
create_time	否	Number	数据连接创建时间, 时间戳。

## 响应参数

状态码: 400

表 7-17 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码: 500

表 7-18 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{
  "dw_config": {
    "clusterId": "353ff458-a560-413e-9b84-33f930cb8057",
    "clusterName": "mrs_3x_autotest_do_not_del",
    "nsUserName": "xxxx",
    "nsPassword": "xxxx",
    "kmsId": "a721616c-9a12-47b1-a805-3cfd3e63cd7",
    "kmsName": "KMS-1111"
  },
  "dw_type": "HBASE",
  "dw_name": "test_hbase_01",
  "agent_id": "91f81a12-75c5-43ce-aab8-7149ecef3b17",
  "agent_name": "cdm-4autotest-nodelete",
  "create_user": "",
  "create_time": 1606879861750,
  "dw_id": "8a94805876212c6201762180f5750003",
  "qualified_name": "hive@192.168.0.4-dayu_cdc_w00591497.62099355b894428e8916573ae635f1f9-workspace-e96e2cddce4340acb61591f8f570059d"
}
```

## 响应示例

### 状态码： 200

Success, 测试创建数据连接成功。

```
{
  "message" : null,
  "is_success" : true
}
```

### 状态码： 400

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

### 状态码： 500

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DebugDataconnectionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DebugDataconnectionRequest request = new DebugDataconnectionRequest();
        ApigDataSourceVo body = new ApigDataSourceVo();
        request.withBody(body);
    }
}
```

```
try {
    DebugDataconnectionResponse response = client.debugDataconnection(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DebugDataconnectionRequest()
        request.body = ApigDataSourceVo(
        )
        response = client.debug_dataconnection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DebugDataconnectionRequest{}
request.Body = &model.ApigDataSourceVo{
}
response, err := client.DebugDataconnection(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，测试创建数据连接成功。
400	BadRequest。
500	Internal Server Error。

### 7.1.4 查询单个数据连接信息

#### 功能介绍

查询单个数据连接信息。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/data-connections/{data\_connection\_id}

表 7-19 路径参数

参数	是否必选	参数类型	描述
data_connection_id	是	String	数据连接id, 可通过 <a href="#">数据连接列表</a> 信息中获取。
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 7-20 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token, 使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码: 200

表 7-21 响应 Body 参数

参数	参数类型	描述
dw_name	String	数据连接名称。
dw_type	String	数据连接类型。
dw_config	Object	连接动态变化配置项, 每种连接略有区别, 建议在界面进行调试。
agent_id	String	CDM集群ID, 获取方法请参见 <a href="#">查询集群列表</a> 。
agent_name	String	代理集群名称, 获取方法请参见 <a href="#">查询集群列表</a> 。
env_type	Integer	0: 开发模式 1: 生产模式, 默认为0。
qualified_name	String	数据连接限定名称。

参数	参数类型	描述
dw_id	String	数据连接id。
create_user	String	数据连接创建者。
create_time	Number	数据连接创建时间，时间戳。
dw_catagory	String	数据连接类别。
update_type	Integer	0: 创建 1: 更新，默认为0。

**状态码： 400**

表 7-22 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

**状态码： 500**

表 7-23 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

**状态码： 400**

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```



## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowDataconnectionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDataconnectionRequest request = new ShowDataconnectionRequest();
        request.withDataConnectionId("{data_connection_id}");
        try {
            ShowDataconnectionResponse response = client.showDataconnection(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowDataconnectionRequest()
        request.data_connection_id = "{data_connection_id}"
        response = client.show_dataconnection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDataconnectionRequest{}
    request.DataConnectionId = "{data_connection_id}"
    response, err := client.ShowDataconnection(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回单个数据连接信息。
400	BadRequest。
500	Internal Server Error。

## 7.1.5 更新数据连接信息

### 功能介绍

更新数据连接信息。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/data-connections/{data\_connection\_id}

表 7-24 路径参数

参数	是否必选	参数类型	描述
data_connection_id	是	String	数据连接id，可通过 <a href="#">数据连接列表</a> 中获取。
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 7-25 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 7-26 请求 Body 参数

参数	是否必选	参数类型	描述
data_source_vos	否	Array of <a href="#">ApigDataSourceVo</a> objects	数据源结构体。

表 7-27 ApigDataSourceVo

参数	是否必选	参数类型	描述
dw_name	是	String	数据连接名称，创建时自定义，编辑时通过 <a href="#">数据连接列表</a> 获取。
dw_type	是	String	数据连接类型，如：DWS、DLI、HIVE、RDS、SparkSQL等，可在console界面查看所有数据连接类型，编辑时通过 <a href="#">数据连接列表</a> 获取。
dw_config	是	Object	连接动态变化配置项，每种连接略有区别，建议在界面进行调试。
agent_id	否	String	CDM集群ID，获取方法请参见 <a href="#">查询集群列表</a> 。
agent_name	否	String	代理集群名称，获取方法请参见 <a href="#">查询集群列表</a> 。
env_type	否	Integer	0：开发模式 1：生产模式，默认为0。
qualified_name	否	String	数据连接限定名称。

参数	是否必选	参数类型	描述
dw_id	否	String	数据连接id, 可通过 <a href="#">数据连接列表</a> 中获取。
create_user	否	String	数据连接创建者。
create_time	否	Number	数据连接创建时间, 时间戳。

## 响应参数

状态码： 400

表 7-28 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-29 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{
  "data_source_vos": [{
    "dw_config": {
      "clusterId": "353ff458-a560-413e-9b84-33f930cb8057",
      "clusterName": "mrs_3x_autotest_do_not_del",
      "dbUserName": "xxxxxx",
      "dbPassword": "xxxxxx",
      "kmsId": "a721616c-9a12-47b1-a805-3cfd3e63cd7",
      "kmsName": "KMS-1111",
      "mrsConnectionType": "agent"
    },
    "env_type": 0,
    "dw_type": "HIVE",
    "dw_name": "test_hive_01",
    "agent_id": "91f81a12-75c5-43ce-aab8-7149ecef3b17",
    "agent_name": "cdm-4autotest-nodelete"
  }]
}
```

## 响应示例

### 状态码： 200

Success, 数据连接信息更新成功。

```
{
  "is_success": true,
  "message": null
}
```

### 状态码： 400

```
{
  "error_code": "DAYU.4402",
  "error_msg": "The operation failed, detail msg {0}."
}
```

### 状态码： 500

```
{
  "error_code": "DAYU.3531",
  "error_msg": "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateDataconnectionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateDataconnectionRequest request = new UpdateDataconnectionRequest();
        request.withDataConnectionId("{data_connection_id}");
        ApigDataSourcesVo body = new ApigDataSourcesVo();
```

```
request.withBody(body);
try {
    UpdateDataconnectionResponse response = client.updateDataconnection(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateDataconnectionRequest()
        request.data_connection_id = "{data_connection_id}"
        request.body = ApigDataSourcesVo(
        )
        response = client.update_dataconnection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDataconnectionRequest{
        request.DataConnectionId = "{data_connection_id}"
        request.Body = &model.ApigDataSourcesVo{
        }
    }
    response, err := client.UpdateDataconnection(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，数据连接信息更新成功。
400	BadRequest。
500	Internal Server Error。

## 7.1.6 删除数据连接

### 功能介绍

删除数据连接。

### 调用方法

请参见[如何调用API](#)。



## URI

DELETE /v1/{project\_id}/data-connections/{data\_connection\_id}

表 7-30 路径参数

参数	是否必选	参数类型	描述
data_connection_id	是	String	数据连接id, 可通过 <a href="#">数据连接列表</a> 中获取。
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 7-31 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token, 使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码: 400

表 7-32 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码: 500

表 7-33 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。

参数	参数类型	描述
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

**状态码： 200**

Success, 数据连接删除成功。

```
{
  "is_success": true,
  "message": null
}
```

**状态码： 400**

```
{
  "error_code": "DAYU.4402",
  "error_msg": "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code": "DAYU.3531",
  "error_msg": "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteDataconnectionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteDataconnectionRequest request = new DeleteDataconnectionRequest();
request.withDataConnectionId("{data_connection_id}");
try {
    DeleteDataconnectionResponse response = client.deleteDataconnection(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteDataconnectionRequest()
        request.data_connection_id = "{data_connection_id}"
        response = client.delete_dataconnection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDataconnectionRequest{}
    request.DataConnectionId = "{data_connection_id}"
    response, err := client.DeleteDataconnection(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，数据连接删除成功。
400	BadRequest。
500	Internal Server Error。

## 7.2 购买实例

## 7.2.1 DataArtsStudio 实例一键购买接口

### 功能介绍

DataArtsStudio实例一键购买接口。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/instances/onekey-purchase

表 7-34 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 7-35 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

表 7-36 请求 Body 参数

参数	是否必选	参数类型	描述
region_id	是	String	通过调用ManageOne平台的IAM服务的“查询区域列表”接口获取响应消息中的区域ID。
commodity_id	否	String	订单Id。
product_id	否	String	产品Id。
period_type	是	Integer	购买周期类型，2：月，3：年。
period_num	是	Integer	购买周期数，大于0。

参数	是否必选	参数类型	描述
availability_zone	是	String	可用区。
vpc_id	是	String	虚拟网卡Id。
security_group_id	是	String	安全组id, 可在当前安全组列表中获取。
net_id	是	String	子网id, 可在当前虚拟私有云列表中获取。
instance_name	是	String	实例名。
eps_id	是	String	企业项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
is_auto_renew	是	Integer	是否续订。
promotion_info	否	String	促销信息。
extesion_package_type	否	String	实例附加增量包类型。
binding_instance_id	否	String	DataArts Studio实例ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
cdm_version	否	String	cdm版本号。
resource_spec_code	是	String	产品规格编码(即实例类型编码), 例如dayu.starter, dayu.basic, dayu.advanced等。可通过查询其他实例信息进行获取。
cloud_service_type	否	String	云服务类型。
resource_type	否	String	资源类型。
tags	否	Array of <a href="#">TmsTagDTO</a> objects	tms标签。

表 7-37 TmsTagDTO

参数	是否必选	参数类型	描述
key	是	String	key值。
value	是	String	value值。

## 响应参数

状态码： 200

表 7-38 响应 Body 参数

参数	参数类型	描述
order_id	String	订单Id。
resource_id	String	实例Id。

状态码： 400

表 7-39 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-40 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{
  "region_id": "xxxxxx",
  "period_type": 2,
  "period_num": 1,
  "availability_zone": "xxxxxx",
  "vpc_id": "000aaa00-8ece-44c0-963c-b6ed0e105c34",
  "net_id": "5a6411c1-1487-420c-8c97-c264a6dae12e",
  "security_group_id": "01263337-b511-4de3-ab2d-f57d9af67d1e",
  "eps_id": "9dc0e37c-c117-48c7-8c6d-ac4b27cfc8a8",
  "instance_name": "DGC-0722",
  "resource_spec_code": "dayu.nb.professional",
  "is_auto_renew": 0,
  "promotion_info": ""
}
```

## 响应示例

状态码： 400

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class PayForDgcOneKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        PayForDgcOneKeyRequest request = new PayForDgcOneKeyRequest();
        OrderReq body = new OrderReq();
        request.withBody(body);
        try {
            PayForDgcOneKeyResponse response = client.payForDgcOneKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```



```
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = PayForDgcOneKeyRequest()  
        request.body = OrderReq(  
        )  
        response = client.pay_for_dgc_one_key(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```

```
client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.PayForDgcOneKeyRequest{
    request.Body = &model.OrderReq{
    }
}
response, err := client.PayForDgcOneKey(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	返回实例Id和订单Id。
400	BadRequest。
500	Internal Server Error。

## 7.3 工作空间管理

### 7.3.1 获取工作空间列表

#### 功能介绍

获取工作空间列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/workspaces/{instance\_id}

表 7-41 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	DataArts Studio实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 7-42 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每次查询的数据连接条数的限制量，默认20。
offset	否	Integer	数据偏移量，默认0。

## 请求参数

表 7-43 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 7-44 响应 Body 参数

参数	参数类型	描述
count	Integer	当前工作空间用户记录数。
total_page	Integer	查询结果总页数。
data	Array of <a href="#">Workspacebody</a> objects	工作空间列表。

表 7-45 Workspacebody

参数	参数类型	描述
bad_record_location_name	String	脏数据OBS路径。
description	String	工作空间描述。
eps_id	String	企业项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_log_location_name	String	作业日志OBS路径。
name	String	工作空间名称。
id	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
is_default	Integer	是否为默认空间，0为私有空间，1为默认空间，2为公共空间。
owner_name	String	创建者名称。
project_id	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
domain_id	String	租户id，可在IAM侧用户信息中查询账号ID。
instance_id	String	DataArts Studio实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
create_time	Number	创建时间。
create_user	String	创建用户名称。
member_num	Integer	当前工作空间成员数量。
update_time	Number	更新时间。
update_user	String	更新用户名称。

状态码： 400

表 7-46 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-47 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

### 状态码： 400

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

### 状态码： 500

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListManagerWorkSpacesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListManagerWorkSpacesRequest request = new ListManagerWorkSpacesRequest();
request.withInstanceId("{instance_id}");
try {
    ListManagerWorkSpacesResponse response = client.listManagerWorkSpaces(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListManagerWorkSpacesRequest()
        request.instance_id = "{instance_id}"
        response = client.list_manager_work_spaces(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```

dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListManagerWorkSpacesRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListManagerWorkSpaces(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回工作空间信息列表。
400	BadRequest。
500	Internal Server Error。

## 7.3.2 创建工作空间

### 功能介绍

创建工作空间。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/workspaces/{instance\_id}

表 7-48 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	DataArts Studio实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 请求参数

表 7-49 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

表 7-50 请求 Body 参数

参数	是否必选	参数类型	描述
bad_record_location_name	否	String	脏数据OBS路径。
description	否	String	工作空间描述。
eps_id	是	String	企业项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。可通过查询同实例下其他工作空间获取，或在console界面获取。
job_log_location_name	否	String	作业日志OBS路径。
name	是	String	工作空间名称。



## 响应参数

状态码： 400

表 7-51 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-52 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{
  "name": "tesfasdfasdf",
  "description": "",
  "eps_id": "0",
  "job_log_location_name": "obs://xxx/ccc/",
  "bad_record_location_name": "obs://aaaaa111/"
}
```

## 响应示例

状态码： 200

Success, 创建工作空间成功。

```
{
  "message": null,
  "is_success": true
}
```

状态码： 400

```
{
  "error_code": "DAYU.4402",
  "error_msg": "The operation failed, detail msg {0}."
}
```

状态码： 500

```
{
  "error_code": "DAYU.3531",
  "error_msg": "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateManagerWorkSpaceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateManagerWorkSpaceRequest request = new CreateManagerWorkSpaceRequest();
        request.withInstanceId("{instance_id}");
        CreateWorkspaceParams body = new CreateWorkspaceParams();
        request.withBody(body);
        try {
            CreateManagerWorkSpaceResponse response = client.createManagerWorkSpace(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateManagerWorkSpaceRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateWorkspaceParams(
        )
        response = client.create_manager_work_space(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateManagerWorkSpaceRequest{
        request.InstanceId = "{instance_id}"
        request.Body = &model.CreateWorkspaceParams{
```

```

}
response, err := client.CreateManagerWorkSpace(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，创建工作空间成功。
400	BadRequest。
500	Internal Server Error。

## 7.3.3 获取单个工作空间信息

### 功能介绍

获取单个工作空间信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/workspaces/{instance\_id}/{workspace\_id}

表 7-53 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	DataArts Studio实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
workspace_id	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 请求参数

表 7-54 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 7-55 响应 Body 参数

参数	参数类型	描述
bad_record_location_name	String	脏数据OBS路径。
description	String	工作空间描述。
eps_id	String	企业项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
job_log_location_name	String	作业日志OBS路径。
name	String	工作空间名称。
id	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
is_default	Integer	是否为默认空间，0为私有空间，1为默认空间，2为公共空间。
owner_name	String	创建者名称。
project_id	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
domain_id	String	租户id，可在IAM侧用户信息中查询账号ID。
instance_id	String	DataArts Studio实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
create_time	Number	创建时间。
create_user	String	创建用户名称。
member_num	Integer	当前工作空间成员数量。
update_time	Number	更新时间。

参数	参数类型	描述
update_user	String	更新用户名称。

状态码： 400

表 7-56 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-57 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

状态码： 400

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

状态码： 500

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowWorkSpaceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowWorkSpaceRequest request = new ShowWorkSpaceRequest();
        request.withInstanceId("{instance_id}");
        request.withWorkspaceId("{workspace_id}");
        try {
            ShowWorkSpaceResponse response = client.showWorkSpace(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
```

```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowWorkSpaceRequest()
    request.instance_id = "{instance_id}"
    request.workspace_id = "{workspace_id}"
    response = client.show_work_space(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowWorkSpaceRequest{}
    request.InstanceId = "{instance_id}"
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ShowWorkSpace(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```



## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回单个工作空间信息。
400	BadRequest。
500	Internal Server Error。

## 7.4 实例管理

### 7.4.1 获取实例列表

#### 功能介绍

获取实例列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/instances

表 7-58 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 7-59 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每次查询的数据连接条数的限制量，默认20。
offset	否	Integer	每次查询的数据连接条数的分页页码，默认0。

## 请求参数

表 7-60 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 7-61 响应 Body 参数

参数	参数类型	描述
billing_check	Boolean	是否需要账单。
count	Integer	返回记录总数。
commodity_orders	Array of ApigCommodityOrder objects	返回实例列表。

表 7-62 ApigCommodityOrder

参数	参数类型	描述
project_id	String	实例所属项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
order_id	String	CBC订单id。
region_id	String	通过调用ManageOne平台的IAM服务的“查询区域列表”接口获取响应消息中的区域ID。
resource_id	String	实例id。
resource_name	String	实例名称。
subscription_id	String	CBC订购id。
resource_type	String	资源类型，hws.resource.type.dayu。

参数	参数类型	描述
resource_spec_code	String	产品规格编码(即实例类型编码)，例如 dayu.starter, dayu.basic, dayu.advanced等。可通过查询其他实例信息进行获取。
product_id	String	CBC产品id, 未安装CBC的环境无需这个值。
order_type	String	订单类型标识符。
charge_type	String	支付选项, 留空。
is_auto_renew	Integer	自动续费标识, 当前实例为按需支付时必填, 0代表不续费, 1代表自动续费。
status	Integer	实例状态, 1未生效2生效中3已删除=退订4保留期=冻结5宽限期6删除中。
vpc_id	String	虚拟私有云id, 可在当前虚拟私有云列表中获取。
security_group_id	String	安全组id, 可在当前安全组列表中获取。
eps_id	String	企业项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
effective_time	Number	生效时间点, 包周期实例有效。
expire_days	String	过期时间天数, 包周期实例有效。
expire_time	Number	过期时间点, 包周期有效。
lock_check_endpoint	String	CBC锁定节点域名。
create_user	String	创建用户。
create_time	Number	创建时间点。
domain_id	String	用户domain id。
is_trial_order	Integer	是否试用订单。
work_space_mode	String	工作空间模式说明。

状态码: 400

表 7-63 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

**状态码： 500**

**表 7-64 响应 Body 参数**

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

**状态码： 400**

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDataArtsStudioInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListDataArtsStudioInstancesRequest request = new ListDataArtsStudioInstancesRequest();
try {
    ListDataArtsStudioInstancesResponse response = client.listDataArtsStudioInstances(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListDataArtsStudioInstancesRequest()
        response = client.list_data_arts_studio_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDataArtsStudioInstancesRequest{}
    response, err := client.ListDataArtsStudioInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回DataArtsStudio实例列表。
400	BadRequest。
500	Internal Server Error。

## 7.5 工作空间用户管理

### 7.5.1 获取工作空间用户角色

#### 功能介绍

获取工作空间用户角色。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/users/role

表 7-65 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 7-66 Query 参数

参数	是否必选	参数类型	描述
instance_id	否	String	DataArts Studio实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
workspace_id	否	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 请求参数

表 7-67 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 7-68 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of <a href="#">ApigRoleVo</a> objects	获取DataArtsStudio工作空间角色列表信息。

表 7-69 ApigRoleVo

参数	参数类型	描述
role_id	String	角色id信息。
role_code	String	角色编码信息。
role_name	String	角色名称。
description	String	角色描述。

状态码： 400

表 7-70 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-71 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

状态码： 400

```
{  
  "error_code" : "DAYU.4402",
```



```
"error_msg" : "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListWorkspaceRolesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListWorkspaceRolesRequest request = new ListWorkspaceRolesRequest();
        try {
            ListWorkspaceRolesResponse response = client.listWorkspaceRoles(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListWorkspaceRolesRequest()
        response = client.list_workspace_roles(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()
{
    request := &model.ListWorkspaceRolesRequest{}
    response, err := client.ListWorkspaceRoles(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回用户角色信息列表。
400	BadRequest。
500	Internal Server Error。

## 7.5.2 编辑工作空间用户或用户组

### 功能介绍

编辑工作空间用户或用户组。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/{workspace\_id}/users/{user\_id}

表 7-72 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
workspace_id	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
user_id	是	String	用户id，可在IAM侧用户信息中查询。

## 请求参数

表 7-73 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <b>Token认证</b> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

表 7-74 请求 Body 参数

参数	是否必选	参数类型	描述
type	是	Integer	用户类型，0：添加用户，1：添加用户组。
user_ids	否	Array of <b>ApiglamUser Dto</b> objects	用户列表信息，从获取工作空间用户信息列表中获取。
groups	否	Array of <b>Group</b> objects	用户组列表信息，从获取工作空间用户信息列表中获取。
roles_ids	是	Array of <b>ApigRole</b> objects	空间角色列表，从获取工作空间用户信息列表中获取。

表 7-75 ApiglamUserDto

参数	是否必选	参数类型	描述
user_id	否	String	用户id，从获取工作空间用户信息列表中获取。
user_name	否	String	用户名，从获取工作空间用户信息列表中获取。
domain_id	否	String	租户id，从获取工作空间用户信息列表中获取。
domain_name	否	String	租户名，从获取工作空间用户信息列表中获取。
is_domain_ow ner	否	Boolean	是否是空间拥有者，从获取工作空间用户信息列表中获取。

表 7-76 Group

参数	是否必选	参数类型	描述
id	否	String	用户组id, 从获取工作空间用户信息列表中获取。
name	否	String	用户组名, 从获取工作空间用户信息列表中获取。

表 7-77 ApigRole

参数	是否必选	参数类型	描述
role_id	否	String	角色id, 可以角色列表信息中获取, r00001: 管理员; r00002: 开发者; r00003: 运维者; r00004: 访客

## 响应参数

状态码: 400

表 7-78 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码: 500

表 7-79 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{
  "user_ids": [ {
    "user_id": "2d032145d82546b5b83cd7a6fd7d0afc",
    "user_name": "ei_dayu_n30011721",
    "domain_id": "0833a5736980d53b0f22c0102ffcbfc0",
    "domain_name": "ei_dayu_y00321344_01",
    "is_domain_owner": "false"
  }
]
```

```
    }],  
    "roles_ids" : [ {  
      "role_id" : "r00003"  
    } ],  
    "type" : 0  
  }  
}
```

## 响应示例

### 状态码： 200

Success, 修改工作空间用户成功。

```
{  
  "message" : null,  
  "is_success" : true  
}
```

### 状态码： 400

```
{  
  "error_code" : "DAYU.4402",  
  "error_msg" : "The operation failed, detail msg {0}."  
}
```

### 状态码： 500

```
{  
  "error_code" : "DAYU.3531",  
  "error_msg" : "Internal server error: {0}"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class UpdateWorkSpaceUserOrGroupSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateWorkSpaceUserOrGroupRequest request = new UpdateWorkSpaceUserOrGroupRequest();
request.withWorkspaceId("{workspace_id}");
request.withUserId("{user_id}");
ApigWorkspaceUserDto body = new ApigWorkspaceUserDto();
request.withBody(body);
try {
    UpdateWorkSpaceUserOrGroupResponse response = client.updateWorkSpaceUserOrGroup(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateWorkSpaceUserOrGroupRequest()
        request.workspace_id = "{workspace_id}"
        request.user_id = "{user_id}"
        request.body = ApigWorkspaceUserDto(
        )
        response = client.update_work_space_user_or_group(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateWorkSpaceUserOrGroupRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.UserId = "{user_id}"
    request.Body = &model.ApigWorkspaceUserDto{
    }
    response, err := client.UpdateWorkSpaceUserOrGroup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，修改工作空间用户成功。
400	BadRequest。
500	Internal Server Error。



## 7.5.3 获取工作空间用户信息

### 功能介绍

获取工作空间用户信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/{workspace\_id}/users

表 7-80 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
workspace_id	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 7-81 Query 参数

参数	是否必选	参数类型	描述
limit	否	String	每次查询的数据连接条数的限制量。
offset	否	String	每次查询的数据连接条数的分页页码。

### 请求参数

表 7-82 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

## 响应参数

状态码： 200

表 7-83 响应 Body 参数

参数	参数类型	描述
count	Integer	当前工作空间用户记录数。
total_page	Integer	查询结果总页数。
data	Array of <a href="#">ApigWorkspaceUserbody</a> objects	工作空间用户列表。

表 7-84 ApigWorkspaceUserbody

参数	参数类型	描述
id	String	记录id, 主键标识。
user_id	String	用户id。
user_name	String	用户名。
domain_id	String	租户id。
domain_name	String	租户名。
display_user_name	String	租户名。
domain_owner	Boolean	是否是空间所有者。
description	String	工作空间用户信息描述。
workspace_id	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
roles	Array of <a href="#">ApigRoleVo</a> objects	角色列表。
create_time	Number	创建时间。
create_user	String	创建人员。
update_time	Number	更新时间。
update_user	String	更新人员。
type	Integer	用户类型, 0用户, 1用户组。

表 7-85 ApigRoleVo

参数	参数类型	描述
role_id	String	角色id信息。
role_code	String	角色编码信息。
role_name	String	角色名称。
description	String	角色描述。

状态码： 400

表 7-86 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-87 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

状态码： 400

```
{  
  "error_code" : "DAYU.4402",  
  "error_msg" : "The operation failed, detail msg {0}."  
}
```

状态码： 500

```
{  
  "error_code" : "DAYU.3531",  
  "error_msg" : "Internal server error: {0}"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListWorkspaceusersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListWorkspaceusersRequest request = new ListWorkspaceusersRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ListWorkspaceusersResponse response = client.listWorkspaceusers(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListWorkspaceusersRequest()
        request.workspace_id = "{workspace_id}"
        response = client.list_workspaceusers(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListWorkspaceusersRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListWorkspaceusers(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回用户信息列表。
400	BadRequest。
500	Internal Server Error。

## 7.5.4 添加工作空间用户

### 功能介绍

添加工作空间用户。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/{workspace\_id}/users

表 7-88 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
workspace_id	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 请求参数

表 7-89 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <b>Token认证</b> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

表 7-90 请求 Body 参数

参数	是否必选	参数类型	描述
type	是	Integer	用户类型，0：添加用户，1：添加用户组。
user_ids	否	Array of <b>ApiglamUser Dto</b> objects	用户列表信息，从获取工作空间用户信息列表中获取。
groups	否	Array of <b>Group</b> objects	用户组列表信息，从获取工作空间用户信息列表中获取。
roles_ids	是	Array of <b>ApigRole</b> objects	空间角色列表，从获取工作空间用户信息列表中获取。

表 7-91 ApiglamUserDto

参数	是否必选	参数类型	描述
user_id	否	String	用户id，从获取工作空间用户信息列表中获取。
user_name	否	String	用户名，从获取工作空间用户信息列表中获取。
domain_id	否	String	租户id，从获取工作空间用户信息列表中获取。
domain_name	否	String	租户名，从获取工作空间用户信息列表中获取。
is_domain_ow ner	否	Boolean	是否是空间拥有者，从获取工作空间用户信息列表中获取。

表 7-92 Group

参数	是否必选	参数类型	描述
id	否	String	用户组id, 从获取工作空间用户信息列表中获取。
name	否	String	用户组名, 从获取工作空间用户信息列表中获取。

表 7-93 ApigRole

参数	是否必选	参数类型	描述
role_id	否	String	角色id, 可以角色列表信息中获取, r00001: 管理员; r00002: 开发者; r00003: 运维者; r00004: 访客

## 响应参数

状态码: 400

表 7-94 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码: 500

表 7-95 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{
  "user_ids": [ {
    "user_id": "2d032145d82546b5b83cd7a6fd7d0afc",
    "user_name": "ei_dayu_n30011721",
    "domain_id": "0833a5736980d53b0f22c0102ffcbfc0",
    "domain_name": "ei_dayu_y00321344_01",
    "is_domain_owner": "false"
  }
]
```



```
    }],  
    "roles_ids" : [ {  
      "role_id" : "r00003"  
    } ],  
    "type" : 0  
  }  
}
```

## 响应示例

### 状态码： 200

Success, 成功添加工作空间用户。

```
{  
  "message" : null,  
  "is_success" : true  
}
```

### 状态码： 400

```
{  
  "error_code" : "DAYU.4402",  
  "error_msg" : "The operation failed, detail msg {0}."  
}
```

### 状态码： 500

```
{  
  "error_code" : "DAYU.3531",  
  "error_msg" : "Internal server error: {0}"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class AddWorkSpaceUsersSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
AddWorkSpaceUsersRequest request = new AddWorkSpaceUsersRequest();
request.withWorkspaceId("{workspace_id}");
ApigWorkspaceUserDto body = new ApigWorkspaceUserDto();
request.withBody(body);
try {
    AddWorkSpaceUsersResponse response = client.addWorkSpaceUsers(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = AddWorkSpaceUsersRequest()
        request.workspace_id = "{workspace_id}"
        request.body = ApigWorkspaceUserDto(
        )
        response = client.add_work_space_users(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddWorkspaceUsersRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.Body = &model.ApigWorkspaceUserDto{
    }
    response, err := client.AddWorkspaceUsers(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，成功添加工作空间用户。
400	BadRequest。
500	Internal Server Error。

## 7.5.5 删除工作空间用户

### 功能介绍

删除工作空间用户。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/{workspace\_id}/delete-users

表 7-96 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
workspace_id	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 请求参数

表 7-97 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

表 7-98 请求 Body 参数

参数	是否必选	参数类型	描述
user_ids	是	Object	用户组列表，从工作空间用户信息列表中获取。

## 响应参数

状态码： 400

表 7-99 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

**状态码： 500**

**表 7-100 响应 Body 参数**

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{  
  "user_ids" : [ "2d032145d82546b5b83cd7a6fd7d0afc" ]  
}
```

## 响应示例

**状态码： 200**

Success, 工作空间用户信息删除成功。

```
{  
  "message" : null,  
  "is_success" : true  
}
```

**状态码： 400**

```
{  
  "error_code" : "DAYU.4402",  
  "error_msg" : "The operation failed, detail msg {0}."  
}
```

**状态码： 500**

```
{  
  "error_code" : "DAYU.3531",  
  "error_msg" : "Internal server error: {0}"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class DeleteWorkspaceusersSolution {  
    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteWorkspaceusersRequest request = new DeleteWorkspaceusersRequest();
request.withWorkspaceId("{workspace_id}");
ApigDelUserParams body = new ApigDelUserParams();
request.withBody(body);
try {
    DeleteWorkspaceusersResponse response = client.deleteWorkspaceusers(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteWorkspaceusersRequest()
        request.workspace_id = "{workspace_id}"
```

```
request.body = ApigDelUserParams(
)
response = client.delete_workspaceusers(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteWorkspaceusersRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.Body = &model.ApigDelUserParams{
    }
    response, err := client.DeleteWorkspaceusers(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 工作空间用户信息删除成功。
400	BadRequest。
500	Internal Server Error。

## 7.6 数据源元数据获取

### 7.6.1 获取数据库列表

#### 功能介绍

获取数据库列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/{connection\_id}/databases

表 7-101 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
connection_id	是	String	数据连接id, 可通过 <a href="#">数据连接列表</a> 中获取。

表 7-102 Query 参数

参数	是否必选	参数类型	描述
limit	否	String	数据条数限制。
offset	否	String	偏移量。



## 请求参数

表 7-103 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必须选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	数据所在空间的id值。

## 响应参数

状态码： 200

表 7-104 响应 Body 参数

参数	参数类型	描述
total_count	Integer	当前数据连接数据库记录数。
dw_id	String	数据连接id。
databases	Array of DatabasesList objects	数据库列表。

表 7-105 DatabasesList

参数	参数类型	描述
database_name	String	数据库名称。
description	String	数据库描述。

状态码： 400

表 7-106 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

**状态码： 500**

表 7-107 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

**状态码： 200**

Success, 返回数据库列表。

```
{
  "dw_id": "f6ce0c0de8f14ba8b7fbf23486b8ec16",
  "databases": [ {
    "database_name": "dlcatalog_2fe5",
    "description": null
  }, {
    "database_name": "dlcatalog_3e24",
    "description": null
  }, {
    "database_name": "dlcatalog_677a",
    "description": null
  }, {
    "database_name": "dlcatalog_86e4",
    "description": null
  }, {
    "database_name": "dlcatalog_ced5",
    "description": null
  } ],
  "total_count": 21
}
```

**状态码： 400**

```
{
  "error_code": "DAYU.4402",
  "error_msg": "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code": "DAYU.3531",
  "error_msg": "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDatabasesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDatabasesRequest request = new ListDatabasesRequest();
        request.withConnectionId("{connection_id}");
        try {
            ListDatabasesResponse response = client.listDatabases(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListDatabasesRequest()
    request.connection_id = "{connection_id}"
    response = client.list_databases(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDatabasesRequest{}
    request.ConnectionId = "{connection_id}"
    response, err := client.ListDatabases(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回数据库列表。
400	BadRequest。
500	Internal Server Error。

## 7.6.2 获取 schemas

### 功能介绍

获取schemas，目前只有DWS和采用postgresql驱动的RDS数据源支持schema，请在调用前确认该数据源是否支持schema字段。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/{connection\_id}/schemas

表 7-108 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
connection_id	是	String	数据连接id，可通过 <a href="#">数据连接列表</a> 中获取。

表 7-109 Query 参数

参数	是否必选	参数类型	描述
database_name	是	String	数据库名称。
limit	否	String	数据条数限制。
offset	否	String	偏移量。

## 请求参数

表 7-110 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	数据所在空间的id值。

## 响应参数

状态码： 200

表 7-111 响应 Body 参数

参数	参数类型	描述
total_count	Integer	当前数据连接schema记录数。
dw_id	String	数据连接id。
database	String	数据库名称。
schemas	Array of SchemasList objects	schema列表。

表 7-112 SchemasList

参数	参数类型	描述
schema_name	String	schema名称。
description	String	schema描述。

状态码： 400

表 7-113 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

**状态码： 500**

**表 7-114 响应 Body 参数**

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

**状态码： 200**

Success, 返回schemas列表。

```
{
  "dw_id" : "9b18c0ad6ef5404caef4e6cbaccdae6f",
  "database" : "postgres",
  "schemas" : [ {
    "schema_name" : "dm_autotest",
    "description" : ""
  }, {
    "schema_name" : "dbadmin",
    "description" : ""
  }, {
    "schema_name" : "public",
    "description" : "gs_roach_stop_backup"
  }, {
    "schema_name" : "schame",
    "description" : ""
  }, {
    "schema_name" : "utl_file",
    "description" : ""
  }, {
    "schema_name" : "utl_raw",
    "description" : ""
  } ],
  "total_count" : 6
}
```

**状态码： 400**

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSchemasSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSchemasRequest request = new ListSchemasRequest();
        request.withConnectionId("{connection_id}");
        try {
            ListSchemasResponse response = client.listSchemas(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```



example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSchemasRequest()
    request.connection_id = "{connection_id}"
    response = client.list_schemas(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSchemasRequest{}
    request.ConnectionId = "{connection_id}"
    response, err := client.ListSchemas(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回schemas列表。
400	BadRequest。
500	Internal Server Error。

## 7.6.3 获取数据源中的表

### 功能介绍

获取数据源中的表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/{connection\_id}/datatables

表 7-115 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
connection_id	是	String	数据连接id，可通过 <a href="#">数据连接列表</a> 中获取。

表 7-116 Query 参数

参数	是否必选	参数类型	描述
database_name	是	String	数据库名称。
table_name	否	String	指定查询表的名称。
limit	否	String	数据条数限制。

参数	是否必选	参数类型	描述
offset	否	String	偏移量。

## 请求参数

表 7-117 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	数据所在空间的id值。

## 响应参数

状态码： 200

表 7-118 响应 Body 参数

参数	参数类型	描述
total_count	Integer	当前数据库中表的记录数。
tables	Array of <a href="#">TablesList</a> objects	表的列表。

表 7-119 TablesList

参数	参数类型	描述
table_name	String	表名称。
table_id	String	数据表的id信息。
table_name_cn	String	表的中文名称。
columns	String	表中字段。
dw_id	String	数据连接id。
dw_name	String	数据连接名称。

参数	参数类型	描述
dw_type	String	数据连接类型。
database_name	String	数据库名称。
schema_name	String	schema名称。
life_cycle	Integer	表的生命周期。
description	String	表的描述。
user_id	String	用户id, 可在IAM侧用户信息中查询。
user_name	String	用户名称。
project_id	String	数据连接id。
create_time	String	表的创建时间。
table_size	Integer	表的大小。
total_count	Integer	当前查询条件下表的总记录数。
is_valid	Integer	表是否合规。
extra_setting	String	表的额外设置。
partitioned	Boolean	是否进行数据分区。

状态码： 400

表 7-120 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-121 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

**状态码： 200**

Success, 返回查询的表的列表。

```
{
  "tables": [ {
    "description": null,
    "table_id": "NativeTable-9b18c0ad6ef5404caef4e6cbaccdcae6f-postgres-dm_autotest-kuaallfuhe01",
    "table_name": "kuaallfuhe01",
    "table_name_cn": null,
    "columns": null,
    "dw_id": "9b18c0ad6ef5404caef4e6cbaccdcae6f",
    "dw_name": "dws_xiaoyu",
    "dw_type": "DWS",
    "database_name": "postgres",
    "schema_name": "dm_autotest",
    "life_cycle": 0,
    "user_id": "ei_dayu_y00321344_01",
    "user_name": null,
    "project_id": null,
    "create_time": null,
    "table_size": 0,
    "total_count": 15,
    "is_valid": 1,
    "extra_setting": null,
    "partitioned": true
  } ],
  "total_count": 15
}
```

**状态码： 400**

```
{
  "error_code": "DAYU.4402",
  "error_msg": "The operation failed, detail msg {0}."
}
```

**状态码： 500**

```
{
  "error_code": "DAYU.3531",
  "error_msg": "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class ListDataTablesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDataTablesRequest request = new ListDataTablesRequest();
        request.withConnectionId("{connection_id}");
        try {
            ListDataTablesResponse response = client.listDataTables(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = ListDataTablesRequest()
    request.connection_id = "{connection_id}"
    response = client.list_data_tables(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDataTablesRequest{}
    request.ConnectionId = "{connection_id}"
    response, err := client.ListDataTables(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回查询的表的列表。

状态码	描述
400	BadRequest。
500	Internal Server Error。

## 7.6.4 获取数据源中表的字段

### 功能介绍

获取数据源中表的字段。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/{connection\_id}/datatables/{table\_id}/columns

表 7-122 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
connection_id	是	String	数据连接id，可通过 <a href="#">数据连接列表</a> 中获取。
table_id	是	String	数据表的id信息。

表 7-123 Query 参数

参数	是否必选	参数类型	描述
limit	否	String	数据条数限制。
offset	否	String	偏移量。



## 请求参数

表 7-124 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	数据所在空间的id值。

## 响应参数

状态码： 200

表 7-125 响应 Body 参数

参数	参数类型	描述
table_id	String	数据表的id信息。
total_count	Integer	当前表中字段记录数。
columns	Array of <a href="#">ColumnsList</a> objects	字段列表。

表 7-126 ColumnsList

参数	参数类型	描述
comment	String	字段注解。
column_name	String	字段名称。
column_type	String	字段类型。
seq_number	Integer	字段的顺序。
primary	Boolean	字段是否为主键。
partition_col	Boolean	是否对字段进行分割。

状态码： 400

表 7-127 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-128 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

无

## 响应示例

状态码： 200

Success, 返回字段列表。

```
{
  "columns": [ {
    "comment": "",
    "column_name": "age",
    "column_type": "integer",
    "seq_number": 2,
    "primary": false,
    "partition_col": false
  }, {
    "comment": "",
    "column_name": "address",
    "column_type": "character varying",
    "seq_number": 3,
    "primary": false,
    "partition_col": false
  }, {
    "comment": "",
    "column_name": "phonenum",
    "column_type": "character varying",
    "seq_number": 4,
    "primary": false,
    "partition_col": false
  } ],
  "total_count": 7,
  "table_id": "NativeTable-9b18c0ad6ef5404caef4e6cbaccdcae6f-postgres-dm_autotest-
  QQQQQQstudents_infoweq"
}
```

状态码： 400

```
{
  "error_code": "DAYU.4402",
}
```

```
"error_msg" : "The operation failed, detail msg {0}."
}
```

**状态码: 500**

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListColumnsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListColumnsRequest request = new ListColumnsRequest();
        request.withConnectionId("{connection_id}");
        request.withTableId("{table_id}");
        try {
            ListColumnsResponse response = client.listColumns(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListColumnsRequest()
        request.connection_id = "{connection_id}"
        request.table_id = "{table_id}"
        response = client.list_columns(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
```

```

        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build()

    request := &model.ListColumnsRequest{}
    request.ConnectionId = "{connection_id}"
    request.TableId = "{table_id}"
    response, err := client.ListColumns(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回字段列表。
400	BadRequest。
500	Internal Server Error。

## 7.7 实例规格变更

### 7.7.1 规格变更接口

#### 功能介绍

规格变更接口。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/{project\_id}/change-resource

表 7-129 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 7-130 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用ManageOne平台的IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

表 7-131 请求 Body 参数

参数	是否必选	参数类型	描述
change_mode	是	Integer	规格变更类型：10：升配；30：降配；40：续费；60：扩容；70：切换操作系统。
resource_id	是	String	DataArts Studio实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
resource_spec_code	是	String	产品规格编码（即实例类型编码），例如dayu.starter，dayu.basic，dayu.advanced等，通过 <a href="#">实例列表</a> 信息进行获取。
product_id	否	String	产品id，通过 <a href="#">实例列表</a> 信息进行获取。
promotion_info	否	String	促销信息，通过 <a href="#">实例列表</a> 信息进行获取。

## 响应参数

状态码： 400

表 7-132 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

状态码： 500

表 7-133 响应 Body 参数

参数	参数类型	描述
error_msg	String	返回的错误信息。
error_code	String	返回的错误码。

## 请求示例

```
{
  "resource_id" : "2b6c183a606f4983b503b6427ba5db30",
  "product_id" : "",
  "resource_spec_code" : "dayu.experience",
  "change_mode" : 30
}
```

## 响应示例

### 状态码： 200

OK, 规格变更请求成功。

```
{
  "message" : null,
  "is_success" : true
}
```

### 状态码： 400

```
{
  "error_code" : "DAYU.4402",
  "error_msg" : "The operation failed, detail msg {0}."
}
```

### 状态码： 500

```
{
  "error_code" : "DAYU.3531",
  "error_msg" : "Internal server error: {0}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ChangeResourceSolution {
    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ChangeResourceRequest request = new ChangeResourceRequest();
ApigChangeResourceReq body = new ApigChangeResourceReq();
request.withBody(body);
try {
    ChangeResourceResponse response = client.changeResource(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ChangeResourceRequest()
        request.body = ApigChangeResourceReq(
        )
```



```
response = client.change_resource(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeResourceRequest{}
    request.Body = &model.ApigChangeResourceReq{
    }
    response, err := client.ChangeResource(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK，规格变更请求成功。
400	BadRequest。

状态码	描述
500	Internal Server Error。

# 8 数据架构 API

## 8.1 概览

### 8.1.1 总览统计信息

#### 功能介绍

总览统计信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/definitions/statistic

表 8-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-3 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-4 data

参数	参数类型	描述
value	<a href="#">StatisticInfo</a> object	value，统一的返回结果的外层数据结构。

表 8-5 StatisticInfo

参数	参数类型	描述
atomic_index	StatisticSchema object	原子指标。
derivative_index	StatisticSchema object	衍生指标。
compound_metric	StatisticSchema object	复合指标。
biz_index	StatisticSchema object	业务指标。
dimension	StatisticSchema object	维度。
condition_group	StatisticSchema object	限定分组。
time_condition	StatisticSchema object	时间限定。
common_condition	StatisticSchema object	通用限定。
dimension_logic_table	StatisticSchema object	维度表。
fact_logic_table	StatisticSchema object	事实表。
aggregation_logic_table	StatisticSchema object	汇总表。
data_standard	StatisticSchema object	数据标准。
table_model	StatisticSchema object	业务表。
lookup_table	StatisticSchema object	码表。
pending_review	Integer	待我审核。
my_applications	Integer	我的申请。

表 8-6 StatisticSchema

参数	参数类型	描述
increase	Integer	本月新增。

参数	参数类型	描述
total	Integer	总量。
standard_coverag e	Double	标准覆盖率。

**状态码：400**

**表 8-7** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-8** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-9** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

总览页面统计信息。

GET [https://{endpoint}/v2/{project\\_id}/design/definitions/statistic](https://{endpoint}/v2/{project_id}/design/definitions/statistic)

## 响应示例

### 状态码：200

Success, 返回data数据是StatisticInfo信息。

```
{
  "data": {
    "value": {
      "atomic_index": {
        "increase": 0,
        "total": 5,
        "standard_coverage": null
      },
      "derivative_index": {
        "increase": 0,
        "total": 8,
        "standard_coverage": null
      },
      "compound_metric": {
        "increase": 0,
        "total": 3,
        "standard_coverage": null
      },
      "biz_index": {
        "increase": 0,
        "total": 1,
        "standard_coverage": null
      },
      "dimension": {
        "increase": 1,
        "total": 22,
        "standard_coverage": null
      },
      "condition_group": null,
      "time_condition": {
        "increase": 0,
        "total": 14,
        "standard_coverage": null
      },
      "common_condition": null,
      "dimension_logic_table": {
        "increase": 1,
        "total": 17,
        "standard_coverage": null
      },
      "fact_logic_table": {
        "increase": 0,
        "total": 7,
        "standard_coverage": null
      },
      "aggregation_logic_table": {
        "increase": 0,
        "total": 12,
        "standard_coverage": null
      },
      "data_standard": {
        "increase": 1,
        "total": 13,
        "standard_coverage": null
      },
      "table_model": {
        "increase": 15,
        "total": 50,
        "standard_coverage": 0.0775
      },
      "lookup_table": {
        "increase": 2,
        "total": 10,

```

```
    "standard_coverage" : null
  },
  "pending_review" : 1,
  "my_applications" : 1
}
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CountOverviewsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
```



```
        .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
CountOverviewsRequest request = new CountOverviewsRequest();
try {
    CountOverviewsResponse response = client.countOverviews(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CountOverviewsRequest()
        response = client.count_overviews(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CountOverviewsRequest{}
    response, err := client.CountOverviews(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是StatisticInfo信息。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.1.2 模型统计信息

### 功能介绍

单个模型中的统计信息。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/design/table-models/statistic

表 8-10 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-11 Query 参数

参数	是否必选	参数类型	描述
model_id	否	String	依据关系建模的模型ID查询统计信息，ID字符串。

## 请求参数

表 8-12 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-13 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-14 data

参数	参数类型	描述
value	<b>TableModelStatisticVO</b> object	value，统一的返回结果的外层数据结构。

表 8-15 TableModelStatisticVO

参数	参数类型	描述
total	<b>StatisticStandardCoverageVO</b> object	逻辑实体数量。
published	<b>StatisticStandardCoverageVO</b> object	已发布逻辑实体数量。
sync_total	<b>SyncStatusStatisticVO</b> object	同步整体状态。
physical_table	<b>SyncStatusStatisticVO</b> object	创建表状态。
technical_asset	<b>SyncStatusStatisticVO</b> object	技术资产同步状态。
business_asset	<b>SyncStatusStatisticVO</b> object	业务资产同步状态。
meta_data_link	<b>SyncStatusStatisticVO</b> object	资产关联状态。
data_quality	<b>SyncStatusStatisticVO</b> object	质量作业状态。

表 8-16 StatisticStandardCoverageVO

参数	参数类型	描述
all_col_num	String	字段总数，ID字符串。
col_num	String	关联标准字段数，ID字符串。
coverage	Double	标准覆盖率。
details	Array of <a href="#">AllTableVO</a> objects	引用表数组。

表 8-17 AllTableVO

参数	参数类型	描述
id	String	l1的ID，ID字符串。
name	String	l1名称。
tb_logic_guid	String	表发布后对应的逻辑实体guid。
quality_id	String	质量ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
tb_guid	String	表发布后对应的物理表guid。
code	String	编码。
create_by	String	创建人。
tenant_id	String	租户ID。
description	String	描述。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"><li>• DRAFT: 草稿</li><li>• PUBLISH_DEVELOPING: 发布待审核</li><li>• PUBLISHED: 已发布</li><li>• OFFLINE_DEVELOPING: 下线待审核</li><li>• OFFLINE: 已下线</li><li>• REJECT: 已驳回</li></ul>

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
db_name	String	数据库名。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。



参数	参数类型	描述
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象ID, 只读, ID字符串。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
standard_count	String	标准数量，只读，ID字符串。
alias	String	别名。
api_id	String	汇总表API ID。
workspace_id	String	工作空间ID。
workspace_name	String	工作空间名称。
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。

参数	参数类型	描述
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"><li>INVALID_TYPE: 无效环境</li><li>DEV_TYPE: 开发环境</li><li>PROD_TYPE: 生产环境</li><li>DEV_PROD_TYPE: 开发生产环境</li></ul>

表 8-18 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-19 SyncStatusStatisticVO

参数	参数类型	描述
success	String	成功数，ID字符串。
failed	String	失败数，ID字符串。
running	String	同步中，ID字符串。
other	String	未同步数，ID字符串。

状态码：400

表 8-20 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-21 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-22 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。



## 请求示例

关系建模统计信息。

```
GET https://{endpoint}/v2/{project_id}/design/table-models/statistic
```

## 响应示例

**状态码：200**

Success, 返回data数据是TableModelStatisticVO信息。

```
{
  "data": {
    "value": {
      "total": {
        "all_col_num": "2",
        "col_num": "20",
        "coverage": 0.15,
        "details": null
      },
      "published": {
        "all_col_num": "2",
        "col_num": "20",
        "coverage": 0.15,
        "details": null
      },
      "sync_total": {
        "success": "1",
        "failed": "1",
        "running": "0",
        "other": "0"
      },
      "physical_table": {
        "success": "2",
        "failed": "0",
        "running": "0",
        "other": "0"
      },
      "technical_asset": {
        "success": "1",
        "failed": "1",
        "running": "0",
        "other": "0"
      },
      "business_asset": {
        "success": "2",
        "failed": "0",
        "running": "0",
        "other": "0"
      },
      "meta_data_link": {
        "success": "1",
        "failed": "1",
        "running": "0",
        "other": "0"
      },
      "data_quality": {
        "success": "2",
        "failed": "0",
        "running": "0",
        "other": "0"
      }
    }
  }
}
```

**状态码：400**

### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CountTableModelsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CountTableModelsRequest request = new CountTableModelsRequest();
        try {
            CountTableModelsResponse response = client.countTableModels(request);
            System.out.println(response.toString());
        }
    }
}
```

```
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CountTableModelsRequest()
        response = client.count_table_models(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```

ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CountTableModelsRequest{}
response, err := client.CountTableModels(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是TableModelStatisticVO信息。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.1.3 关系建模统计信息

### 功能介绍

关系建模页面，外层的统计信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/models/statistic

表 8-23 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-24 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-25 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-26 data

参数	参数类型	描述
value	<b>AllModelStatisticVO</b> object	value, 统一的返回结果的外层数据结构。

表 8-27 AllModelStatisticVO

参数	参数类型	描述
frequent	Array of <b>ModelStatisticVO</b> objects	是否为常用。
top	Array of <b>ModelStatisticVO</b> objects	首层模型。
logic	Array of <b>ModelStatisticVO</b> objects	逻辑模型。
physical	Array of <b>ModelStatisticVO</b> objects	物理模型。
dwr	<b>ModelStatisticVO</b> object	DWR数据报告层。
dm	<b>ModelStatisticVO</b> object	DM数据整合层。

表 8-28 ModelStatisticVO

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值:</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构 (批量修改主题使用)</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>SUBJECT: 主题</li> <li>SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>SUMMARY_INDEX: 汇总表指标属性</li> <li>SUMMARY_TIME: 汇总表时间周期属性</li> <li>TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>TABLE_MODEL_LOGIC: 逻辑实体</li> <li>TABLE_TYPE: 表类型</li> <li>TAG: 标签</li> <li>TIME_CONDITION: 时间限定</li> </ul>
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>SDI: 贴源数据层</li> <li>DWI: 数据整合层</li> <li>DWR: 数据报告层</li> <li>DM: 数据集市层</li> </ul>
db	Integer	数据库。
tb	Integer	数据表。
tb_published	Integer	已发布的数据表。
fd	Integer	字段。
fd_published	Integer	已发布的字段。
st	Double	标准覆盖率。
st_published	Double	已发布的标准覆盖率。
model	<b>WorkspaceVO</b> object	模型信息。

表 8-29 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。



参数	参数类型	描述
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-30 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-31 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-32 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

关系建模统计信息。

GET https://{endpoint}/v2/{project\_id}/design/models/statistic

## 响应示例

状态码：200

Success，返回data数据是AllModelStatisticVO信息。

```
{
  "data": {
    "value": {
      "frequent": null,
      "top": [ {
        "biz_type": "TABLE_MODEL",
        "level": "SDI",
```

```

"db" : 2,
"tb" : 14,
"tb_published" : 10,
"fd" : 31,
"fd_published" : 20,
"st" : 0.06451612903225806,
"st_published" : 0.1,
"model" : {
  "id" : "1183882892003127296",
  "name" : "test_prefix_001",
  "description" : "",
  "is_physical" : true,
  "frequent" : false,
  "top" : true,
  "level" : "SDI",
  "dw_type" : "DWS",
  "create_time" : "2023-12-11T21:27:57+08:00",
  "update_time" : "2024-04-08T19:18:05+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "type" : "THIRD_NF",
  "biz_catalog_ids" : null,
  "databases" : null,
  "table_model_prefix" : "test_"
}
}, {
  "biz_type" : "TABLE_MODEL",
  "level" : null,
  "db" : 0,
  "tb" : 0,
  "tb_published" : 0,
  "fd" : 0,
  "fd_published" : 0,
  "st" : 0.0,
  "st_published" : 0.0,
  "model" : {
    "id" : "1230823879853875200",
    "name" : "LOGIC_B",
    "description" : "",
    "is_physical" : false,
    "frequent" : false,
    "top" : true,
    "level" : null,
    "dw_type" : "UNSPECIFIED",
    "create_time" : "2024-04-19T10:14:41+08:00",
    "update_time" : "2024-04-19T10:14:41+08:00",
    "create_by" : "test_uesr",
    "update_by" : "test_uesr",
    "type" : "THIRD_NF",
    "biz_catalog_ids" : null,
    "databases" : null,
    "table_model_prefix" : ""
  }
}
}],
"logic" : [ {
  "biz_type" : "TABLE_MODEL",
  "level" : null,
  "db" : 0,
  "tb" : 0,
  "tb_published" : 0,
  "fd" : 0,
  "fd_published" : 0,
  "st" : 0.0,
  "st_published" : 0.0,
  "model" : {
    "id" : "1230823879853875200",
    "name" : "LOGIC_B",
    "description" : "",
    "is_physical" : false,

```

```
"frequent" : false,
"top" : true,
"level" : null,
"dw_type" : "UNSPECIFIED",
"create_time" : "2024-04-19T10:14:41+08:00",
"update_time" : "2024-04-19T10:14:41+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"type" : "THIRD_NF",
"biz_catalog_ids" : null,
"databases" : null,
"table_model_prefix" : ""
}
}, {
  "biz_type" : "TABLE_MODEL",
  "level" : null,
  "db" : 0,
  "tb" : 0,
  "tb_published" : 0,
  "fd" : 0,
  "fd_published" : 0,
  "st" : 0.0,
  "st_published" : 0.0,
  "model" : {
    "id" : "1230823841513799680",
    "name" : "LOGIC_A",
    "description" : "",
    "is_physical" : false,
    "frequent" : false,
    "top" : false,
    "level" : null,
    "dw_type" : "UNSPECIFIED",
    "create_time" : "2024-04-19T10:14:31+08:00",
    "update_time" : "2024-04-19T10:14:31+08:00",
    "create_by" : "test_uesr",
    "update_by" : "test_uesr",
    "type" : "THIRD_NF",
    "biz_catalog_ids" : null,
    "databases" : null,
    "table_model_prefix" : ""
  }
}
}, {
  "biz_type" : "TABLE_MODEL",
  "level" : null,
  "db" : 0,
  "tb" : 4,
  "tb_published" : 3,
  "fd" : 12,
  "fd_published" : 9,
  "st" : 0.16666666666666666,
  "st_published" : 0.11111111111111111,
  "model" : {
    "id" : "1229432454457262080",
    "name" : "test_czh",
    "description" : "",
    "is_physical" : false,
    "frequent" : false,
    "top" : false,
    "level" : null,
    "dw_type" : "UNSPECIFIED",
    "create_time" : "2024-04-15T14:05:39+08:00",
    "update_time" : "2024-04-15T14:05:39+08:00",
    "create_by" : "test_uesr",
    "update_by" : "test_uesr",
    "type" : "THIRD_NF",
    "biz_catalog_ids" : null,
    "databases" : null,
    "table_model_prefix" : ""
  }
}
}
```

```
}, {
  "biz_type": "TABLE_MODEL",
  "level": null,
  "db": 0,
  "tb": 11,
  "tb_published": 10,
  "fd": 25,
  "fd_published": 23,
  "st": 0.04,
  "st_published": 0.043478260869565216,
  "model": {
    "id": "1169318954880122880",
    "name": "logic_czh",
    "description": "",
    "is_physical": false,
    "frequent": false,
    "top": false,
    "level": null,
    "dw_type": "UNSPECIFIED",
    "create_time": "2023-11-01T16:56:04+08:00",
    "update_time": "2023-11-01T16:56:04+08:00",
    "create_by": "test_uesr",
    "update_by": "test_uesr",
    "type": "THIRD_NF",
    "biz_catalog_ids": null,
    "databases": null,
    "table_model_prefix": null
  }
}],
"physical": [ {
  "biz_type": "TABLE_MODEL",
  "level": null,
  "db": 1,
  "tb": 9,
  "tb_published": 7,
  "fd": 34,
  "fd_published": 20,
  "st": 0.08823529411764706,
  "st_published": 0.1,
  "model": {
    "id": "1204448983481765888",
    "name": "test_dws",
    "description": "",
    "is_physical": true,
    "frequent": false,
    "top": false,
    "level": null,
    "dw_type": "DWS",
    "create_time": "2024-02-06T15:30:15+08:00",
    "update_time": "2024-02-06T15:30:15+08:00",
    "create_by": "test_uesr",
    "update_by": "test_uesr",
    "type": "THIRD_NF",
    "biz_catalog_ids": null,
    "databases": null,
    "table_model_prefix": ""
  }
}],
  "biz_type": "TABLE_MODEL",
  "level": "SDI",
  "db": 2,
  "tb": 14,
  "tb_published": 10,
  "fd": 31,
  "fd_published": 20,
  "st": 0.06451612903225806,
  "st_published": 0.1,
  "model": {
    "id": "1183882892003127296",
```

```

"name": "test_prefix_001",
"description": "",
"is_physical": true,
"frequent": false,
"top": true,
"level": "SDI",
"dw_type": "DWS",
"create_time": "2023-12-11T21:27:57+08:00",
"update_time": "2024-04-08T19:18:05+08:00",
"create_by": "test_uesr",
"update_by": "test_uesr",
"type": "THIRD_NF",
"biz_catalog_ids": null,
"databases": null,
"table_model_prefix": "test_"
}
}, {
"biz_type": "TABLE_MODEL",
"level": null,
"db": 0,
"tb": 2,
"tb_published": 2,
"fd": 4,
"fd_published": 4,
"st": 0.0,
"st_published": 0.0,
"model": {
"id": "1178728794341699584",
"name": "dws_czh",
"description": "",
"is_physical": true,
"frequent": false,
"top": false,
"level": null,
"dw_type": "DWS",
"create_time": "2023-11-27T16:07:24+08:00",
"update_time": "2024-04-08T19:13:13+08:00",
"create_by": "test_uesr",
"update_by": "test_uesr",
"type": "THIRD_NF",
"biz_catalog_ids": null,
"databases": null,
"table_model_prefix": "test_"
}
}, {
"biz_type": "TABLE_MODEL",
"level": null,
"db": 2,
"tb": 10,
"tb_published": 8,
"fd": 36,
"fd_published": 32,
"st": 0.08333333333333333,
"st_published": 0.09375,
"model": {
"id": "1169319480548048896",
"name": "mrs_hive_czh",
"description": "",
"is_physical": true,
"frequent": false,
"top": false,
"level": null,
"dw_type": "MRS_HIVE",
"create_time": "2023-11-01T16:58:09+08:00",
"update_time": "2023-11-01T16:58:09+08:00",
"create_by": "test_uesr",
"update_by": "test_uesr",
"type": "THIRD_NF",
"biz_catalog_ids": null,

```

```

        "databases" : null,
        "table_model_prefix" : null
    }
}],
"dwr" : {
    "biz_type" : "DIMENSION",
    "level" : null,
    "db" : 5,
    "tb" : 24,
    "tb_published" : 21,
    "fd" : 387,
    "fd_published" : 378,
    "st" : 0.0,
    "st_published" : 0.0,
    "model" : null
},
"dm" : {
    "biz_type" : "AGGREGATION_LOGIC_TABLE",
    "level" : null,
    "db" : 2,
    "tb" : 12,
    "tb_published" : 7,
    "fd" : 77,
    "fd_published" : 35,
    "st" : 0.0,
    "st_published" : 0.0,
    "model" : null
}
}
}
}

```

#### 状态码：400

##### BadRequest

```

{
    "error_code" : "DS.60xx",
    "error_msg" : "The user request is illegal."
}

```

#### 状态码：401

##### Unauthorized

```

{
    "error_code" : "DS.60xx",
    "error_msg" : "User authentication failed."
}

```

#### 状态码：403

##### Forbidden

```

{
    "error_code" : "DS.60xx",
    "error_msg" : "The user does not have permission to call this API."
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;

```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CountAllModelsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CountAllModelsRequest request = new CountAllModelsRequest();
        try {
            CountAllModelsResponse response = client.countAllModels(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```



```
client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CountAllModelsRequest()
    response = client.count_all_models(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CountAllModelsRequest{}
    response, err := client.CountAllModels(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是AllModelStatisticVO信息。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.1.4 标准覆盖率统计信息

### 功能介绍

查看某个数据标准在所有模型字段中的覆盖率，即使用该标准的字段占总字段的百分比。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/definitions/statistic/{id}

表 8-33 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID, ID字符串。

表 8-34 Query 参数

参数	是否必选	参数类型	描述
biz_type	否	String	按业务类型查询，可选业务类型有：ATOMIC_INDEX（原子指标）、DERIVATIVE_INDEX（衍生指标）、DIMENSION（维度）、TIME_CONDITION（时间限定）、DIMENSION_LOGIC_TABLE（维度表）、FACT_LOGIC_TABLE（事实表）、AGGREGATION_LOGIC_TABLE（汇总表）、TABLE_MODEL（关系建模表）、CODE_TABLE（码表）、STANDARD_ELEMENT（数据标准）、BIZ_METRIC（业务指标）、COMPOUND_METRIC（复合指标）、SUBJECT（主题）、ATOMIC_METRIC（原子指标（新））、DERIVED_METRIC（衍生指标（新））、COMPOSITE_METRIC（复合指标（新））。

## 请求参数

表 8-35 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-36 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-37 data

参数	参数类型	描述
value	<b>StatisticStandardCoverageVO</b> object	value, 统一的返回结果的外层数据结构。

表 8-38 StatisticStandardCoverageVO

参数	参数类型	描述
all_col_num	String	字段总数, ID字符串。
col_num	String	关联标准字段数, ID字符串。
coverage	Double	标准覆盖率。
details	Array of <b>AllTableVO</b> objects	引用表数组。

表 8-39 AllTableVO

参数	参数类型	描述
id	String	l1的ID, ID字符串。

参数	参数类型	描述
name	String	l1名称。
tb_logic_guid	String	表发布后对应的逻辑实体guid。
quality_id	String	质量ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
tb_guid	String	表发布后对应的物理表guid。
code	String	编码。
create_by	String	创建人。
tenant_id	String	租户ID。
description	String	描述。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
db_name	String	数据库名。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。

参数	参数类型	描述
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象ID, 只读, ID字符串。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
standard_count	String	标准数量，只读，ID字符串。
alias	String	别名。
api_id	String	汇总表API ID。
workspace_id	String	工作空间ID。
workspace_name	String	工作空间名称。
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。

参数	参数类型	描述
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"><li>INVALID_TYPE: 无效环境</li><li>DEV_TYPE: 开发环境</li><li>PROD_TYPE: 生产环境</li><li>DEV_PROD_TYPE: 开发生产环境</li></ul>

表 8-40 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-41 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-42 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-43 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查看ID为1186622399320305665的数据标准的表属性覆盖率。

```
GET https://{endpoint}/v2/{project_id}/design/definitions/statistic/1186622399320305665?
biz_type=STANDARD_ELEMENT
```

## 响应示例

**状态码：200**

Success，返回data数据是StatisticStandardCoverageVO信息。

```
{
  "data": {
    "value": {
      "all_col_num": "621",
      "col_num": "1",
      "coverage": 0.001610305958132045,
```

```

"details" : [ {
  "id" : "1202300665725100032",
  "name" : "test_czh_secrety_0055",
  "tb_logic_guid" : "163b03fa-44c8-4eb1-9844-fdec203325b0",
  "quality_id" : null,
  "reversed" : null,
  "partition_conf" : null,
  "dirty_out_switch" : false,
  "dirty_out_database" : "",
  "dirty_out_prefix" : "",
  "dirty_out_suffix" : "",
  "tb_guid" : null,
  "code" : "test_czh_secrety_005",
  "create_by" : "test_uesr",
  "tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "description" : "无",
  "status" : "PUBLISHED",
  "biz_type" : "TABLE_MODEL",
  "create_time" : "2024-01-31T17:13:36+08:00",
  "update_time" : "2024-04-08T15:50:39+08:00",
  "db_name" : null,
  "dw_type" : "DWS",
  "queue_name" : null,
  "schema" : null,
  "l1" : null,
  "l2" : null,
  "l3" : null,
  "l1_id" : null,
  "l2_id" : null,
  "l3_id" : "1194966710038474752",
  "new_biz" : null,
  "physical_table" : "UPDATE_FAILED",
  "dev_physical_table" : "NO_NEED",
  "technical_asset" : "CREATE_FAILED",
  "business_asset" : "NO_NEED",
  "meta_data_link" : "UPDATE_FAILED",
  "data_quality" : "NO_NEED",
  "dlf_task" : "NO_NEED",
  "materialization" : null,
  "publish_to_dlm" : null,
  "summary_status" : "SUMMARY_FAILED",
  "standard_count" : "1",
  "alias" : null,
  "api_id" : "0",
  "workspace_id" : "9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "workspace_name" : "test_czh_simple",
  "dev_version" : null,
  "prod_version" : "1202300758293389312",
  "dev_version_name" : null,
  "prod_version_name" : null,
  "env_type" : "PROD_TYPE"
} ]
}
}
}

```

**状态码: 400**

BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

**状态码: 401**

Unauthorized



```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CountStandardsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CountStandardsRequest request = new CountStandardsRequest();
        request.withId("{id}");
        try {
            CountStandardsResponse response = client.countStandards(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CountStandardsRequest()  
        request.id = "{id}"  
        response = client.count_standards(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CountStandardsRequest{}
request.Id = "{id}"
response, err := client.CountStandards(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是StatisticStandardCoverageVO信息。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.2 信息架构接口

### 8.2.1 查询多种类型的表信息

#### 功能介绍

从信息架构中查询多种类型的表信息，包括逻辑实体、物理表、维度表、事实表、汇总表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/all-tables

表 8-44 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-45 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
sync_status	否	String	同步状态枚举。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 同步中</li> <li>• NO_NEED: 未同步</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> </ul>

参数	是否必选	参数类型	描述
sync_key	否	Array of strings	同步任务类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• BUSINESS_ASSET: 同步业务资产</li> <li>• DATA_QUALITY: 创建质量作业</li> <li>• TECHNICAL_ASSET: 同步技术资产</li> <li>• META_DATA_LINK: 资产关联</li> <li>• PHYSICAL_TABLE: 创建表（生产环境）</li> <li>• DEV_PHYSICAL_TABLE: 创建表（开发环境）</li> <li>• DLF_TASK: 创建数据开发作业</li> <li>• MATERIALIZATION: 数值落库（码表）</li> <li>• PUBLISH_TO_DLM: 发布数据服务API</li> <li>• SUMMARY_STATUS: 整体状态</li> </ul>
biz_catalog_id	否	String	所属的业务分层的ID。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

参数	是否必选	参数类型	描述
biz_catalog_id_list	否	Array of strings	所属主题的ID列表，ID字符串。
biz_type_list	是	Array of strings	查询的表类型，必填。 枚举值： <ul style="list-style-type: none"> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_MODEL: 物理表</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> </ul>

## 请求参数

表 8-46 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-47 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-48 data

参数	参数类型	描述
value	<b>value</b> object	value, 统一的返回结果的外层数据结构。

表 8-49 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>AllTableVO</b> objects	查询到的审批单对象 ( AllTableVO ) 数组。

表 8-50 AllTableVO

参数	参数类型	描述
id	String	l1的ID, ID字符串。
name	String	l1名称。
tb_logic_guid	String	表发布后对应的逻辑实体guid。
quality_id	String	质量ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_databas e	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
tb_guid	String	表发布后对应的物理表guid。
code	String	编码。

参数	参数类型	描述
create_by	String	创建人。
tenant_id	String	租户ID。
description	String	描述。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
db_name	String	数据库名。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。

参数	参数类型	描述
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象ID, 只读, ID字符串。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
standard_count	String	标准数量，只读，ID字符串。
alias	String	别名。
api_id	String	汇总表API ID。
workspace_id	String	工作空间ID。
workspace_name	String	工作空间名称。
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。

参数	参数类型	描述
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"><li>INVALID_TYPE: 无效环境</li><li>DEV_TYPE: 开发环境</li><li>PROD_TYPE: 生产环境</li><li>DEV_PROD_TYPE: 开发生产环境</li></ul>

表 8-51 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-52 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-53 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-54 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

从信息架构中查询逻辑实体、物理表、维度表、事实表、汇总表的前十条数据。

```
GET https://{endpoint}/v2/{project_id}/design/all-tables?
offset=1&limit=10&biz_type_list=TABLE_MODEL_LOGIC&biz_type_list=TABLE_MODEL&biz_type_list=DIMENSI
ON_LOGIC_TABLE&biz_type_list=FACT_LOGIC_TABLE&biz_type_list=AGGREGATION_LOGIC_TABLE&advOnInit
Flg=true
```

## 响应示例

状态码：200

Success，返回data数据是AllTableVO列表。

```
{
  "data" : {
    "value" : {
      "total" : 86,
```

```

"records": [ {
  "id": "1229920251379118081",
  "name": "test_czh_0416",
  "tb_logic_guid": "",
  "quality_id": "1230938508982857728",
  "reversed": null,
  "partition_conf": null,
  "dirty_out_switch": false,
  "dirty_out_database": "",
  "dirty_out_prefix": "",
  "dirty_out_suffix": "",
  "tb_guid": null,
  "code": "test_czh_0416",
  "create_by": "test_uesr",
  "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "description": "无",
  "status": "OFFLINE",
  "biz_type": "TABLE_MODEL",
  "create_time": "2024-04-16T22:23:59+08:00",
  "update_time": "2024-04-19T17:55:49+08:00",
  "db_name": "bi",
  "dw_type": "DWS",
  "queue_name": null,
  "schema": null,
  "l1": "test_czh_catalog",
  "l2": null,
  "l3": null,
  "l1_id": "1194966710038474752",
  "l2_id": null,
  "l3_id": null,
  "new_biz": null,
  "physical_table": "CREATE_FAILED",
  "dev_physical_table": "NO_NEED",
  "technical_asset": "CREATE_FAILED",
  "business_asset": "CREATE_SUCCESS",
  "meta_data_link": "UPDATE_FAILED",
  "data_quality": "CREATE_SUCCESS",
  "dlf_task": "NO_NEED",
  "materialization": null,
  "publish_to_dlm": null,
  "summary_status": "SUMMARY_FAILED",
  "standard_count": null,
  "alias": null,
  "api_id": "0",
  "workspace_id": null,
  "workspace_name": null,
  "dev_version": null,
  "prod_version": null,
  "dev_version_name": null,
  "prod_version_name": null,
  "env_type": "PROD_TYPE"
}, {
  "id": "1229812609512062976",
  "name": "test_czh_sl_003",
  "tb_logic_guid": "86ddea4c-0dda-4a7e-8ca1-a3ddce64ad97",
  "quality_id": null,
  "reversed": null,
  "partition_conf": null,
  "dirty_out_switch": false,
  "dirty_out_database": "",
  "dirty_out_prefix": "",
  "dirty_out_suffix": "",
  "tb_guid": null,
  "code": "test_czh_sl_003",
  "create_by": "test_uesr",
  "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "description": "无",
  "status": "PUBLISHED",
  "biz_type": "TABLE_MODEL",

```

```

"create_time" : "2024-04-16T15:16:15+08:00",
"update_time" : "2024-04-17T17:07:55+08:00",
"db_name" : null,
"dw_type" : "UNSPECIFIED",
"queue_name" : null,
"schema" : null,
"l1" : "test_czh2",
"l2" : null,
"l3" : null,
"l1_id" : "1170064724101115904",
"l2_id" : null,
"l3_id" : null,
"new_biz" : null,
"physical_table" : "NO_NEED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "NO_NEED",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "NO_NEED",
"data_quality" : "CREATE_FAILED",
"dfl_task" : "NO_NEED",
"materialization" : null,
"publish_to_dlm" : null,
"summary_status" : "SUMMARY_SUCCESS",
"standard_count" : null,
"alias" : null,
"api_id" : "0",
"workspace_id" : null,
"workspace_name" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
}, {
"id" : "1229812231022243840",
"name" : "test_czh_sl_002",
"tb_logic_guid" : "bafa9927-9f2b-4197-b8fa-cc94338b707b",
"quality_id" : null,
"reversed" : null,
"partition_conf" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"tb_guid" : null,
"code" : "test_czh_sl_002",
"create_by" : "test_uesr",
"tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "无",
"status" : "PUBLISHED",
"biz_type" : "TABLE_MODEL",
"create_time" : "2024-04-16T15:14:45+08:00",
"update_time" : "2024-04-16T15:14:52+08:00",
"db_name" : null,
"dw_type" : "UNSPECIFIED",
"queue_name" : null,
"schema" : null,
"l1" : "test_czh2",
"l2" : null,
"l3" : null,
"l1_id" : "1170064724101115904",
"l2_id" : null,
"l3_id" : null,
"new_biz" : null,
"physical_table" : "NO_NEED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "NO_NEED",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "NO_NEED",

```

```

"data_quality": "NO_NEED",
"dlf_task": "NO_NEED",
"materialization": null,
"publish_to_dlm": null,
"summary_status": "SUMMARY_SUCCESS",
"standard_count": null,
"alias": null,
"api_id": "0",
"workspace_id": null,
"workspace_name": null,
"dev_version": null,
"prod_version": null,
"dev_version_name": null,
"prod_version_name": null,
"env_type": "PROD_TYPE"
}, {
"id": "1229805021919621120",
"name": "test_czh_sl_001",
"tb_logic_guid": "cec6e323-ec00-457a-aaee-9ec931326bc7",
"quality_id": null,
"reversed": null,
"partition_conf": null,
"dirty_out_switch": false,
"dirty_out_database": "",
"dirty_out_prefix": "",
"dirty_out_suffix": "",
"tb_guid": null,
"code": "test_czh_sl_001",
"create_by": "test_uesr",
"tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description": "无",
"status": "PUBLISHED",
"biz_type": "TABLE_MODEL",
"create_time": "2024-04-16T14:46:06+08:00",
"update_time": "2024-04-16T14:50:34+08:00",
"db_name": null,
"dw_type": "UNSPECIFIED",
"queue_name": null,
"schema": null,
"l1": "test_czh_catalog",
"l2": null,
"l3": null,
"l1_id": "1194966710038474752",
"l2_id": null,
"l3_id": null,
"new_biz": null,
"physical_table": "NO_NEED",
"dev_physical_table": "NO_NEED",
"technical_asset": "NO_NEED",
"business_asset": "CREATE_SUCCESS",
"meta_data_link": "NO_NEED",
"data_quality": "NO_NEED",
"dlf_task": "NO_NEED",
"materialization": null,
"publish_to_dlm": null,
"summary_status": "SUMMARY_SUCCESS",
"standard_count": null,
"alias": null,
"api_id": "0",
"workspace_id": null,
"workspace_name": null,
"dev_version": null,
"prod_version": null,
"dev_version_name": null,
"prod_version_name": null,
"env_type": "PROD_TYPE"
}, {
"id": "1226973789422813184",
"name": "test_0408_004",

```

```

"tb_logic_guid" : "40322535-e903-44e1-a0c0-f79573fc09e6",
"quality_id" : "1227637237613944833",
"reversed" : null,
"partition_conf" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"tb_guid" : null,
"code" : "test_0408_004",
"create_by" : "test_uesr",
"tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "无",
"status" : "PUBLISHED",
"biz_type" : "TABLE_MODEL",
"create_time" : "2024-04-08T19:15:47+08:00",
"update_time" : "2024-04-10T15:12:17+08:00",
"db_name" : null,
"dw_type" : "DWS",
"queue_name" : null,
"schema" : null,
"l1" : "测试主题",
"l2" : null,
"l3" : null,
"l1_id" : "1170064658061799424",
"l2_id" : null,
"l3_id" : null,
"new_biz" : null,
"physical_table" : "CREATE_FAILED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "CREATE_SUCCESS",
"dlf_task" : "NO_NEED",
"materialization" : null,
"publish_to_dlm" : null,
"summary_status" : "SUMMARY_FAILED",
"standard_count" : null,
"alias" : null,
"api_id" : "0",
"workspace_id" : null,
"workspace_name" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
}, {
"id" : "1226973648695504896",
"name" : "test_0408_003",
"tb_logic_guid" : "d6b610a5-de9f-43f6-b3ef-aa4bc23ae766",
"quality_id" : null,
"reversed" : null,
"partition_conf" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"tb_guid" : null,
"code" : "test_0408_003",
"create_by" : "test_uesr",
"tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "无",
"status" : "PUBLISHED",
"biz_type" : "TABLE_MODEL",
"create_time" : "2024-04-08T19:15:14+08:00",
"update_time" : "2024-04-08T19:18:36+08:00",
"db_name" : null,

```

```

"dw_type" : "DWS",
"queue_name" : null,
"schema" : null,
"l1" : "我的测试",
"l2" : null,
"l3" : null,
"l1_id" : "1178033294286176256",
"l2_id" : null,
"l3_id" : null,
"new_biz" : null,
"physical_table" : "CREATE_FAILED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "UPDATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "NO_NEED",
"dlf_task" : "NO_NEED",
"materialization" : null,
"publish_to_dlm" : null,
"summary_status" : "SUMMARY_FAILED",
"standard_count" : null,
"alias" : null,
"api_id" : "0",
"workspace_id" : null,
"workspace_name" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
}, {
"id" : "1226923543347003392",
"name" : "test_0408_002",
"tb_logic_guid" : "b11128db-ba65-40b6-99a5-6103106fac79",
"quality_id" : null,
"reversed" : null,
"partition_conf" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"tb_guid" : null,
"code" : "test_0408_002",
"create_by" : "test_uesr",
"tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "无",
"status" : "PUBLISHED",
"biz_type" : "TABLE_MODEL",
"create_time" : "2024-04-08T15:56:08+08:00",
"update_time" : "2024-04-08T19:14:09+08:00",
"db_name" : null,
"dw_type" : "DWS",
"queue_name" : null,
"schema" : null,
"l1" : "test_czh2",
"l2" : null,
"l3" : null,
"l1_id" : "1170064724101115904",
"l2_id" : null,
"l3_id" : null,
"new_biz" : null,
"physical_table" : "CREATE_FAILED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "UPDATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "NO_NEED",
"dlf_task" : "NO_NEED",
"materialization" : null,

```

```

"publish_to_dlm" : null,
"summary_status" : "SUMMARY_FAILED",
"standard_count" : null,
"alias" : null,
"api_id" : "0",
"workspace_id" : null,
"workspace_name" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
}, {
  "id" : "1226923406239399936",
  "name" : "test_0408",
  "tb_logic_guid" : "",
  "quality_id" : null,
  "reversed" : null,
  "partition_conf" : null,
  "dirty_out_switch" : false,
  "dirty_out_database" : "",
  "dirty_out_prefix" : "",
  "dirty_out_suffix" : "",
  "tb_guid" : null,
  "code" : "test_0408",
  "create_by" : "test_uesr",
  "tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "description" : "无",
  "status" : "DRAFT",
  "biz_type" : "TABLE_MODEL",
  "create_time" : "2024-04-08T15:55:35+08:00",
  "update_time" : "2024-04-19T17:22:02+08:00",
  "db_name" : null,
  "dw_type" : "DWS",
  "queue_name" : null,
  "schema" : null,
  "l1" : "test_import",
  "l2" : null,
  "l3" : null,
  "l1_id" : "1193142949320474624",
  "l2_id" : null,
  "l3_id" : null,
  "new_biz" : null,
  "physical_table" : "CREATE_FAILED",
  "dev_physical_table" : "NO_NEED",
  "technical_asset" : "CREATE_FAILED",
  "business_asset" : "CREATE_SUCCESS",
  "meta_data_link" : "UPDATE_FAILED",
  "data_quality" : "NO_NEED",
  "dlf_task" : "NO_NEED",
  "materialization" : null,
  "publish_to_dlm" : null,
  "summary_status" : "SUMMARY_FAILED",
  "standard_count" : null,
  "alias" : null,
  "api_id" : "0",
  "workspace_id" : null,
  "workspace_name" : null,
  "dev_version" : null,
  "prod_version" : null,
  "dev_version_name" : null,
  "prod_version_name" : null,
  "env_type" : "PROD_TYPE"
}, {
  "id" : "1222930979413499904",
  "name" : "test_czh_0328",
  "tb_logic_guid" : "712edcdd-3bba-4077-b182-4a1f17e5bcd8",
  "quality_id" : null,
  "reversed" : null,

```



```

"partition_conf" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"tb_guid" : null,
"code" : "dim_test_czh_0328",
"create_by" : "test_uesr",
"tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "无",
"status" : "PUBLISHED",
"biz_type" : "DIMENSION_LOGIC_TABLE",
"create_time" : "2024-03-28T15:31:06+08:00",
"update_time" : "2024-03-28T15:31:10+08:00",
"db_name" : "bi",
"dw_type" : "DWS",
"queue_name" : null,
"schema" : null,
"l1" : "test_czh_catalog",
"l2" : null,
"l3" : null,
"l1_id" : "1194966710038474752",
"l2_id" : null,
"l3_id" : null,
"new_biz" : null,
"physical_table" : "CREATE_FAILED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "NO_NEED",
"dlf_task" : "NO_NEED",
"materialization" : null,
"publish_to_dlm" : null,
"summary_status" : "SUMMARY_FAILED",
"standard_count" : null,
"alias" : "",
"api_id" : "0",
"workspace_id" : null,
"workspace_name" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
}, {
"id" : "1222853173564289024",
"name" : "test_czh_0328",
"tb_logic_guid" : "",
"quality_id" : null,
"reversed" : null,
"partition_conf" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"tb_guid" : null,
"code" : "test_czh_0328",
"create_by" : "test_uesr",
"tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "无",
"status" : "OFFLINE",
"biz_type" : "TABLE_MODEL",
"create_time" : "2024-03-28T10:21:56+08:00",
"update_time" : "2024-04-19T17:55:50+08:00",
"db_name" : "bi",
"dw_type" : "DWS",
"queue_name" : null,
"schema" : null,

```

```
"l1" : "test_czh_catalog",
"l2" : null,
"l3" : null,
"l1_id" : "1194966710038474752",
"l2_id" : null,
"l3_id" : null,
"new_biz" : null,
"physical_table" : "CREATE_FAILED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "NO_NEED",
"dlf_task" : "NO_NEED",
"materialization" : null,
"publish_to_dlm" : null,
"summary_status" : "SUMMARY_FAILED",
"standard_count" : null,
"alias" : null,
"api_id" : "0",
"workspace_id" : null,
"workspace_name" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
  }}
}
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListAllTablesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAllTablesRequest request = new ListAllTablesRequest();
        try {
            ListAllTablesResponse response = client.listAllTables(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListAllTablesRequest()
    response = client.list_all_tables(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAllTablesRequest{}
    response, err := client.ListAllTables(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是AllTableVO列表。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.3 数据标准接口

### 8.3.1 获取数据标准集合

#### 功能介绍

根据查询条件分页获取数据标准集合，按修改时间降序排序。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/standards

表 8-55 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-56 Query 参数

参数	是否必选	参数类型	描述
directory_id	否	String	目录ID。获取该目录下的数据标准，如果有子目录，同时获取所有子目录的数据标准。
name_ch	否	String	按名称精确查询。
name_en	否	String	按编码精确查询。

参数	是否必选	参数类型	描述
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-57 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-58 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回数据标准集合。

表 8-59 data

参数	参数类型	描述
value	<b>value</b> object	返回的数据标准详细信息。

表 8-60 value

参数	参数类型	描述
records	Array of <b>StandElementValueVOList</b> objects	StandElementValueVO数组。
total	Integer	数据标准的总数。

表 8-61 StandElementValueVOList

参数	参数类型	描述
values	Array of <b>StandElementValueVO</b> objects	属性信息。
id	String	数据标准的ID, ID字符串。
directory_id	String	标准所属目录, ID字符串。
directory_path	String	目录树。
row_id	String	标准行的ID, ID字符串。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-62 StandElementValueVO

参数	参数类型	描述
fd_name	String	属性名称。
fd_value	String	属性值。
fd_id	String	属性定义的ID，ID字符串。
directory_id	String	标准所属目录，ID字符串。
row_id	String	标准所属行，ID字符串。
id	String	数据标准的ID，ID字符串。



参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-63 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-64 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

**表 8-65 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-66 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-67 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

**表 8-68 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询目录ID为1193957835831218179及其子目录下的前十条数据标准的集合。

```
GET https://{endpoint}/v2/{project_id}/design/standards?  
offset=1&limit=10&directory_id=1193957835831218179
```

## 响应示例

**状态码: 200**

Success, 返回data数据是该目录及其子目录下的全部数据标准集合, 以及总数。

```
{  
  "data": {  
    "value": {  
      "total": 1,  
      "records": [ {  
        "values": [ {  
          "fd_name": "nameCh",  
          "fd_value": "test_czh_bug_0110",  
          "fd_id": "1169308633595617280",  
          "directory_id": "1193957835831218179",  
          "row_id": "1195028291904507904",  
          "id": "1195028291917090826",  
          "status": "DRAFT",  
          "create_by": "test_uesr",  
          "update_by": "test_uesr",  
          "create_time": "2024-01-11T15:35:47+08:00",  
          "update_time": "2024-01-12T17:33:42+08:00"  
        }, {  
          "fd_name": "nameEn",  
          "fd_value": "DS000005",  
          "fd_id": "1169308634786799616",  
          "directory_id": "1193957835831218179",  
          "row_id": "1195028291904507904",  
          "id": "1195028291917090827",  
          "status": "DRAFT",  
          "create_by": "test_uesr",  
          "update_by": "test_uesr",  
          "create_time": "2024-01-11T15:35:47+08:00",  
          "update_time": "2024-01-12T17:33:42+08:00"  
        }, {  
          "fd_name": "dataType",  
          "fd_value": "STRING",  
          "fd_id": "1169308634803576832",  
          "directory_id": "1193957835831218179",  
          "row_id": "1195028291904507904",  
          "id": "1195028291917090828",  
          "status": "DRAFT",  
          "create_by": "test_uesr",  
          "update_by": "test_uesr",  
          "create_time": "2024-01-11T15:35:47+08:00",  
          "update_time": "2024-01-12T17:33:42+08:00"  
        }, {  
          "fd_name": "dataLength",  
          "fd_value": "",  
          "fd_id": "1169308634824548352",  
          "directory_id": "1193957835831218179",  
          "row_id": "1195028291904507904",  
          "id": "1195028291917090816",  
          "status": "DRAFT",  
          "create_by": "test_uesr",  
          "update_by": "test_uesr",  
          "create_time": "2024-01-11T15:35:47+08:00",  
          "update_time": "2024-01-12T17:33:42+08:00"  
        }, {  
          "fd_name": "hasAllowValueList",  
          "fd_value": "false",  
          "fd_id": "1169308634841325568",  
          "directory_id": "1193957835831218179",  
          "row_id": "1195028291904507904",  
          "id": "1195028291917090817",  
          "status": "DRAFT",  
          "create_by": "test_uesr",  
          "update_by": "test_uesr",  
          "create_time": "2024-01-11T15:35:47+08:00",  
          "update_time": "2024-01-12T17:33:42+08:00"  
        }  
      ]  
    }  
  }  
}
```



```

"status": "DRAFT",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-01-11T15:35:47+08:00",
"update_time": "2024-01-12T17:33:42+08:00"
}, {
"fd_name": "allowList",
"fd_value": "",
"fd_id": "1169308634862297088",
"directory_id": "1193957835831218179",
"row_id": "1195028291904507904",
"id": "1195028291917090818",
"status": "DRAFT",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-01-11T15:35:47+08:00",
"update_time": "2024-01-12T17:33:42+08:00"
}, {
"fd_name": "referCodeTable",
"fd_value": "",
"fd_id": "1169308634879074304",
"directory_id": "1193957835831218179",
"row_id": "1195028291904507904",
"id": "1195028291917090819",
"status": "DRAFT",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-01-11T15:35:47+08:00",
"update_time": "2024-01-12T17:33:42+08:00"
}, {
"fd_name": "codeStandColumn",
"fd_value": "",
"fd_id": "1169308634895851520",
"directory_id": "1193957835831218179",
"row_id": "1195028291904507904",
"id": "1195028291917090820",
"status": "DRAFT",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-01-11T15:35:47+08:00",
"update_time": "2024-01-12T17:33:42+08:00"
}, {
"fd_name": "ruleOwner",
"fd_value": "",
"fd_id": "1169308634933600256",
"directory_id": "1193957835831218179",
"row_id": "1195028291904507904",
"id": "1195028291917090821",
"status": "DRAFT",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-01-11T15:35:47+08:00",
"update_time": "2024-01-12T17:33:42+08:00"
}, {
"fd_name": "dataMonitorOwner",
"fd_value": "",
"fd_id": "1169308634950377472",
"directory_id": "1193957835831218179",
"row_id": "1195028291904507904",
"id": "1195028291917090822",
"status": "DRAFT",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-01-11T15:35:47+08:00",
"update_time": "2024-01-12T17:33:42+08:00"
}, {
"fd_name": "standardLevel",
"fd_value": "",
"fd_id": "1169308634971348992",

```

```

"directory_id" : "1193957835831218179",
"row_id" : "1195028291904507904",
"id" : "1195028291917090823",
"status" : "DRAFT",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-01-11T15:35:47+08:00",
"update_time" : "2024-01-12T17:33:42+08:00"
}, {
  "fd_name" : "description",
  "fd_value" : "",
  "fd_id" : "1169308634988126208",
  "directory_id" : "1193957835831218179",
  "row_id" : "1195028291904507904",
  "id" : "1195028291917090824",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-01-11T15:35:47+08:00",
  "update_time" : "2024-01-12T17:33:42+08:00"
}, {
  "fd_name" : "englishName",
  "fd_value" : "",
  "fd_id" : "1194674245716848640",
  "directory_id" : "1193957835831218179",
  "row_id" : "1195028291904507904",
  "id" : "1195028291917090825",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-01-11T15:35:47+08:00",
  "update_time" : "2024-01-12T17:33:42+08:00"
}, {
  "fd_name" : "啊啊啊啊",
  "fd_value" : "you can you up",
  "fd_id" : "1194674245767180288",
  "directory_id" : "1193957835831218179",
  "row_id" : "1195028291904507904",
  "id" : "1195409867876298763",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-01-12T16:52:02+08:00",
  "update_time" : "2024-01-12T17:33:42+08:00"
}, {
  "fd_name" : "不不不",
  "fd_value" : "no can no bb",
  "fd_id" : "1194674684046856192",
  "directory_id" : "1193957835831218179",
  "row_id" : "1195028291904507904",
  "id" : "1195409867876298764",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-01-12T16:52:02+08:00",
  "update_time" : "2024-01-12T17:33:42+08:00"
} ],
"id" : "1195028291904507904",
"directory_id" : "1193957835831218179",
"directory_path" : "test_czh2",
"row_id" : "1195028291904507904",
"status" : "DRAFT",
"approval_info" : null,
"new_biz" : null,
"from_public" : false,
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-01-11T15:35:47+08:00",
"update_time" : "2024-01-12T17:33:42+08:00"

```

```
    }  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListAllStandardsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListAllStandardsRequest request = new ListAllStandardsRequest();  
        try {  
            ListAllStandardsResponse response = client.listAllStandards(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAllStandardsRequest()
        response = client.list_all_standards(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAllStandardsRequest{}
    response, err := client.ListAllStandards(request)
    if err == nil {
```

```

    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是该目录及其子目录下的全部数据标准集合，以及总数。
400	失败，用户请求非法。
401	失败，用户身份认证失败。
403	失败，用户无权限调用此API。
404	失败，用户请求API不存在。

## 8.3.2 创建数据标准

### 功能介绍

创建数据标准。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/standards

表 8-69 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-70 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-71 请求 Body 参数

参数	是否必选	参数类型	描述
values	是	Array of <a href="#">StandElementValueVO</a> objects	属性信息。
id	否	String	数据标准的ID，ID字符串。
directory_id	是	String	标准所属目录，ID字符串。
directory_path	否	String	目录树。
row_id	否	String	标准行的ID，ID字符串。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManageVO object	业务版本管理，只读。
from_public	否	Boolean	是否来自公共层。
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-72 StandElementValueVO

参数	是否必选	参数类型	描述
fd_name	是	String	属性名称。
fd_value	否	String	属性值。
fd_id	否	String	属性定义的ID，ID字符串。
directory_id	否	String	标准所属目录，ID字符串。
row_id	否	String	标准所属行，ID字符串。
id	否	String	数据标准的ID，ID字符串。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-73 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。



参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-74 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

## 响应参数

状态码：200

表 8-75 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	数据标准创建返回data结果。

表 8-76 data

参数	参数类型	描述
value	<b>StandElementValueVOList</b> object	属性值列表。

表 8-77 StandElementValueVOList

参数	参数类型	描述
values	Array of <b>StandElementValueVO</b> objects	属性信息。
id	String	数据标准的ID，ID字符串。
directory_id	String	标准所属目录，ID字符串。
directory_path	String	目录树。
row_id	String	标准行的ID，ID字符串。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-78 StandElementValueVO

参数	参数类型	描述
fd_name	String	属性名称。
fd_value	String	属性值。
fd_id	String	属性定义的ID，ID字符串。
directory_id	String	标准所属目录，ID字符串。
row_id	String	标准所属行，ID字符串。
id	String	数据标准的ID，ID字符串。



参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-79 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-80 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

**表 8-81 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-82 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-83 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

**表 8-84 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据参数创建数据标准。

POST https://{endpoint}/v2/{project\_id}/design/standards

```
{
  "id" : 0,
  "directory_id" : "1228084836548411397",
  "values" : [ {
    "fd_name" : "nameCh",
    "fd_value" : "test_create_data_standard"
  }, {
    "fd_name" : "nameEn",
    "fd_value" : ""
  }, {
    "fd_name" : "englishName",
    "fd_value" : "test_create_data_standard"
  }, {
    "fd_name" : "dataType",
    "fd_value" : "STRING"
  }, {
    "fd_name" : "dataLength",
    "fd_value" : "1,256"
  }, {
    "fd_name" : "hasAllowValueList",
    "fd_value" : false
  }, {
    "fd_name" : "allowList",
    "fd_value" : ""
  }, {
    "fd_name" : "referCodeTable",
    "fd_value" : ""
  }, {
    "fd_name" : "codeStandColumn",
    "fd_value" : ""
  }, {
    "fd_name" : "dqcRule"
  }, {
    "fd_name" : "ruleOwner",
    "fd_value" : "zty_tess"
  }, {
    "fd_name" : "dataMonitorOwner",
    "fd_value" : "zty_tess"
  }, {
    "fd_name" : "standardLevel",
    "fd_value" : ""
  }, {
    "fd_name" : "description",
    "fd_value" : "这是一个测试用例。"
  } ]
}
```

## 响应示例

状态码：200

Success

```
{
  "data" : {
    "value" : {
      "values" : [ {
        "fd_name" : "nameCh",
        "fd_value" : "test_create_data_standard",
        "fd_id" : "1020622096960831488",
        "directory_id" : "1228084836548411397",
        "row_id" : "1228084992354222081",
        "id" : "1228084992358416384",
        "status" : "DRAFT",
        "create_by" : "test_uesr",
        "update_by" : "test_uesr",
        "create_time" : "2024-04-11T20:51:18.894+08:00",

```



```

"update_time" : "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name" : "nameEn",
  "fd_value" : "DS006001",
  "fd_id" : "1020622096985997312",
  "directory_id" : "1228084836548411397",
  "row_id" : "1228084992354222081",
  "id" : "1228084992358416385",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-04-11T20:51:18.894+08:00",
  "update_time" : "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name" : "englishName",
  "fd_value" : "test_create_data_standard",
  "fd_id" : "1185628711836360704",
  "directory_id" : "1228084836548411397",
  "row_id" : "1228084992354222081",
  "id" : "1228084992358416386",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-04-11T20:51:18.894+08:00",
  "update_time" : "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name" : "dataType",
  "fd_value" : "STRING",
  "fd_id" : "1020622097006968832",
  "directory_id" : "1228084836548411397",
  "row_id" : "1228084992354222081",
  "id" : "1228084992358416387",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-04-11T20:51:18.894+08:00",
  "update_time" : "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name" : "dataLength",
  "fd_value" : "1,256",
  "fd_id" : "1020622097032134656",
  "directory_id" : "1228084836548411397",
  "row_id" : "1228084992354222081",
  "id" : "1228084992358416388",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-04-11T20:51:18.894+08:00",
  "update_time" : "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name" : "hasAllowValueList",
  "fd_value" : "false",
  "fd_id" : "1020622097048911872",
  "directory_id" : "1228084836548411397",
  "row_id" : "1228084992354222081",
  "id" : "1228084992358416389",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-04-11T20:51:18.894+08:00",
  "update_time" : "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name" : "allowList",
  "fd_value" : "",
  "fd_id" : "1020622097065689088",
  "directory_id" : "1228084836548411397",
  "row_id" : "1228084992354222081",
  "id" : "1228084992358416390",
  "status" : "DRAFT",

```

```

"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-04-11T20:51:18.894+08:00",
"update_time": "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name": "referCodeTable",
  "fd_value": "",
  "fd_id": "1020622097086660608",
  "directory_id": "1228084836548411397",
  "row_id": "1228084992354222081",
  "id": "1228084992358416391",
  "status": "DRAFT",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": "2024-04-11T20:51:18.894+08:00",
  "update_time": "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name": "codeStandColumn",
  "fd_value": "",
  "fd_id": "1020622097103437824",
  "directory_id": "1228084836548411397",
  "row_id": "1228084992354222081",
  "id": "1228084992358416392",
  "status": "DRAFT",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": "2024-04-11T20:51:18.894+08:00",
  "update_time": "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name": "dqcRule",
  "fd_value": null,
  "fd_id": "1020622097124409344",
  "directory_id": "1228084836548411397",
  "row_id": "1228084992354222081",
  "id": "1228084992358416393",
  "status": "DRAFT",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": "2024-04-11T20:51:18.894+08:00",
  "update_time": "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name": "ruleOwner",
  "fd_value": "zty_tess",
  "fd_id": "1020622097141186560",
  "directory_id": "1228084836548411397",
  "row_id": "1228084992354222081",
  "id": "1228084992358416394",
  "status": "DRAFT",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": "2024-04-11T20:51:18.894+08:00",
  "update_time": "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name": "dataMonitorOwner",
  "fd_value": "zty_tess",
  "fd_id": "1020622097162158080",
  "directory_id": "1228084836548411397",
  "row_id": "1228084992354222081",
  "id": "1228084992358416395",
  "status": "DRAFT",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": "2024-04-11T20:51:18.894+08:00",
  "update_time": "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name": "standardLevel",
  "fd_value": "",
  "fd_id": "1020622097178935296",
  "directory_id": "1228084836548411397",

```

```
"row_id" : "1228084992354222081",
"id" : "1228084992358416396",
"status" : "DRAFT",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-04-11T20:51:18.894+08:00",
"update_time" : "2024-04-11T20:51:18.894+08:00"
}, {
  "fd_name" : "description",
  "fd_value" : "这是一个测试用例。",
  "fd_id" : "1020622097195712512",
  "directory_id" : "1228084836548411397",
  "row_id" : "1228084992354222081",
  "id" : "1228084992358416397",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-04-11T20:51:18.894+08:00",
  "update_time" : "2024-04-11T20:51:18.894+08:00"
}],
"id" : "1228084992354222081",
"directory_id" : "1228084836548411397",
"directory_path" : null,
"row_id" : null,
"status" : "DRAFT",
"approval_info" : null,
"new_biz" : null,
"from_public" : false,
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-04-11T20:51:18.893+08:00",
"update_time" : "2024-04-11T20:51:18.893+08:00"
}
}
```

### 状态码：400

#### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

#### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

#### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

### 状态码：404

#### Not Found

```
{
  "error_code" : "DS.60xx",
```

```
"error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据参数创建数据标准。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CreateStandardSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateStandardRequest request = new CreateStandardRequest();  
        StandElementValueVOList body = new StandElementValueVOList();  
        List<StandElementValueVO> listbodyValues = new ArrayList<>();  
        listbodyValues.add(  
            new StandElementValueVO()  
                .withFdName("nameCh")  
                .withFdValue("test_create_data_standard")  
        );  
        listbodyValues.add(  
            new StandElementValueVO()  
                .withFdName("nameEn")  
                .withFdValue("")  
        );  
        listbodyValues.add(  
            new StandElementValueVO()  
                .withFdName("englishName")  
                .withFdValue("test_create_data_standard")  
        );  
        listbodyValues.add(  
            new StandElementValueVO()  
                .withFdName("dataType")
```

```
        .withFdValue("STRING")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("dataLength")
            .withFdValue("1,256")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("hasAllowValueList")
            .withFdValue("false")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("allowList")
            .withFdValue("")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("referCodeTable")
            .withFdValue("")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("codeStandColumn")
            .withFdValue("")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("dqcRule")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("ruleOwner")
            .withFdValue("zty_tess")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("dataMonitorOwner")
            .withFdValue("zty_tess")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("standardLevel")
            .withFdValue("")
    );
    listbodyValues.add(
        new StandElementValueVO()
            .withFdName("description")
            .withFdValue("这是一个测试用例。")
    );
    body.withDirectoryId("1228084836548411397");
    body.withId("0");
    body.withValues(listbodyValues);
    request.withBody(body);
    try {
        CreateStandardResponse response = client.createStandard(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

```
}  
}
```

## Python

根据参数创建数据标准。

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateStandardRequest()  
        listValuesbody = [  
            StandElementValueVO(  
                fd_name="nameCh",  
                fd_value="test_create_data_standard"  
            ),  
            StandElementValueVO(  
                fd_name="nameEn",  
                fd_value=""  
            ),  
            StandElementValueVO(  
                fd_name="englishName",  
                fd_value="test_create_data_standard"  
            ),  
            StandElementValueVO(  
                fd_name="dataType",  
                fd_value="STRING"  
            ),  
            StandElementValueVO(  
                fd_name="dataLength",  
                fd_value="1,256"  
            ),  
            StandElementValueVO(  
                fd_name="hasAllowValueList",  
                fd_value="false"  
            ),  
            StandElementValueVO(  
                fd_name="allowList",  
                fd_value=""  
            ),  
            StandElementValueVO(  
                fd_name="referCodeTable",  
                fd_value=""  
            ),  
            StandElementValueVO(  
                fd_name="codeStandColumn",
```

```
        fd_value=""
    ),
    StandElementValueVO(
        fd_name="dqcRule"
    ),
    StandElementValueVO(
        fd_name="ruleOwner",
        fd_value="zty_tess"
    ),
    StandElementValueVO(
        fd_name="dataMonitorOwner",
        fd_value="zty_tess"
    ),
    StandElementValueVO(
        fd_name="standardLevel",
        fd_value=""
    ),
    StandElementValueVO(
        fd_name="description",
        fd_value="这是一个测试用例。"
    )
]
request.body = StandElementValueVOList(
    directory_id="1228084836548411397",
    id="0",
    values=listValuesbody
)
response = client.create_standard(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据参数创建数据标准。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.CreateStandardRequest{  
    fdValueValues:= "test_create_data_standard"  
    fdValueValues1:= ""  
    fdValueValues2:= "test_create_data_standard"  
    fdValueValues3:= "STRING"  
    fdValueValues4:= "1,256"  
    fdValueValues5:= "false"  
    fdValueValues6:= ""  
    fdValueValues7:= ""  
    fdValueValues8:= ""  
    fdValueValues9:= "zty_tess"  
    fdValueValues10:= "zty_tess"  
    fdValueValues11:= ""  
    fdValueValues12:= "这是一个测试用例。"  
    var listValuesbody = []model.StandElementValueVo{  
        {  
            FdName: "nameCh",  
            FdValue: &fdValueValues,  
        },  
        {  
            FdName: "nameEn",  
            FdValue: &fdValueValues1,  
        },  
        {  
            FdName: "englishName",  
            FdValue: &fdValueValues2,  
        },  
        {  
            FdName: "dataType",  
            FdValue: &fdValueValues3,  
        },  
        {  
            FdName: "dataLength",  
            FdValue: &fdValueValues4,  
        },  
        {  
            FdName: "hasAllowValueList",  
            FdValue: &fdValueValues5,  
        },  
        {  
            FdName: "allowList",  
            FdValue: &fdValueValues6,  
        },  
        {  
            FdName: "referCodeTable",  
            FdValue: &fdValueValues7,  
        },  
        {  
            FdName: "codeStandColumn",  
            FdValue: &fdValueValues8,  
        },  
        {  
            FdName: "dqcRule",  
        },  
        {  
            FdName: "ruleOwner",  
            FdValue: &fdValueValues9,  
        },  
        {  
            FdName: "dataMonitorOwner",  
            FdValue: &fdValueValues10,  
        },  
        {  
            FdName: "standardLevel",  
            FdValue: &fdValueValues11,  
        },  
    }  
}
```



```
        FdName: "description",
        FdValue: &fdValueValues12,
    },
}
idStandElementValueVoList:= "0"
request.Body = &model.StandElementValueVoList{
    DirectoryId: "1228084836548411397",
    Id: &idStandElementValueVoList,
    Values: listValuesbody,
}
response, err := client.CreateStandard(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

### 8.3.3 删除数据标准

#### 功能介绍

删除数据标准。

#### 调用方法

请参见[如何调用API](#)。

#### URI

DELETE /v2/{project\_id}/design/standards

表 8-85 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-86 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-87 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-88 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-89 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-90 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-91 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-92 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-93 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

删除数据标准。

```
DELETE https://{endpoint}/v2/{project_id}/design/standards
{
  "ids" : [ "1228086003290537985", "1228084992354222081" ]
}
```

## 响应示例

状态码：200

Success，返回data数据是删除个数，注意：传的id列表是rowid，删除整个数据标准，而不是某个属性。

```
{
  "data" : {
    "value" : 0
  }
}
```

状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

删除数据标准。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteStandardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteStandardRequest request = new DeleteStandardRequest();
```

```
IdsParam body = new IdsParam();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1228086003290537985");
listbodyIds.add("1228084992354222081");
body.withIds(listbodyIds);
request.withBody(body);
try {
    DeleteStandardResponse response = client.deleteStandard(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

删除数据标准。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteStandardRequest()
        listIdsbody = [
            "1228086003290537985",
            "1228084992354222081"
        ]
        request.body = IdsParam(
            ids=listIdsbody
        )
        response = client.delete_standard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

删除数据标准。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteStandardRequest{}
    var listIdsbody = []string{
        "1228086003290537985",
        "1228084992354222081",
    }
    request.Body = &model.IdsParam{
        Ids: listIdsbody,
    }
    response, err := client.DeleteStandard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数，注意：传的id列表是rowid，删除整个数据标准，而不是某个属性。
400	BadRequest

状态码	描述
401	Unauthorized
403	Forbidden
404	Not Found

## 8.3.4 修改数据标准

### 功能介绍

修改数据标准。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/standards/{id}

表 8-94 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

### 请求参数

表 8-95 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。



参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-96 请求 Body 参数

参数	是否必选	参数类型	描述
values	是	Array of <a href="#">StandElementValueVO</a> objects	属性信息。
id	否	String	数据标准的ID，ID字符串。
directory_id	是	String	标准所属目录，ID字符串。
directory_path	否	String	目录树。
row_id	否	String	标准行的ID，ID字符串。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	否	<a href="#">ApprovalVO</a> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	<a href="#">BizVersionManageVO</a> object	业务版本管理，只读。

参数	是否必选	参数类型	描述
from_public	否	Boolean	是否来自公共层。
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-97 StandElementValueVO

参数	是否必选	参数类型	描述
fd_name	是	String	属性名称。
fd_value	否	String	属性值。
fd_id	否	String	属性定义的ID, ID字符串。
directory_id	否	String	标准所属目录, ID字符串。
row_id	否	String	标准所属行, ID字符串。
id	否	String	数据标准的ID, ID字符串。
status	否	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-98 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID, ID字符串。
tenant_id	否	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。

参数	是否必选	参数类型	描述
name_en	否	String	业务英文名。
biz_id	否	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-99 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>



参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

## 响应参数

状态码：200

表 8-100 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	数据标准更新返回data结果。

表 8-101 data

参数	参数类型	描述
value	<b>StandElementValueVOList</b> object	属性值列表。

表 8-102 StandElementValueVOList

参数	参数类型	描述
values	Array of <b>StandElementValueVO</b> objects	属性信息。
id	String	数据标准的ID，ID字符串。
directory_id	String	标准所属目录，ID字符串。
directory_path	String	目录树。
row_id	String	标准行的ID，ID字符串。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-103 StandElementValueVO

参数	参数类型	描述
fd_name	String	属性名称。
fd_value	String	属性值。
fd_id	String	属性定义的ID，ID字符串。
directory_id	String	标准所属目录，ID字符串。
row_id	String	标准所属行，ID字符串。
id	String	数据标准的ID，ID字符串。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-104 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-105 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-106 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-107 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-108 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-109 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

修改ID为1082235079767502848的数据标准信息。

PUT https://{endpoint}/v2/{project\_id}/design/standards/1082235079767502848

```
{
  "id": "1082235079767502848",
  "directory_id": "1082230138994827264",
  "values": [ {
    "id": "1082235079767502860",
    "fd_name": "nameCh",
    "fd_value": "测试8M数据标准导入111449"
  }, {
    "id": "1082235079767502861",
    "fd_name": "nameEn",
    "fd_value": "qn_test_1450_update"
  }, {
    "fd_name": "englishName",
    "fd_value": ""
  }, {
    "id": "1082235079767502862",
    "fd_name": "dataType",
    "fd_value": "STRING"
  }, {
    "id": "1082235079767502849",
    "fd_name": "dataLength",
    "fd_value": "100"
  }, {
    "id": "1082235079767502850",
    "fd_name": "hasAllowValueList",
    "fd_value": false
  }, {
    "id": "1082235079767502851",
    "fd_name": "allowList",
    "fd_value": ""
  }, {
    "id": "1082235079767502852",
    "fd_name": "referCodeTable",
    "fd_value": ""
  }, {
    "id": "1082235079767502853",
    "fd_name": "codeStandColumn"
  }, {
    "fd_name": "dqRule"
  }, {
    "id": "1082235079767502854",
    "fd_name": "ruleOwner",
    "fd_value": "zty_tess"
  }, {
    "id": "1082235079767502855",
    "fd_name": "dataMonitorOwner",
    "fd_value": "zty_tess"
  }, {
    "id": "1082235079767502856",
    "fd_name": "standardLevel"
  }, {
    "id": "1082235079767502857",
    "fd_name": "description",
    "fd_value": "测试8M数据标准导入的描述"
  }
  ]
}
```

## 响应示例

**状态码: 200**

Success

```
{
  "data": {
    "value": {
      "values": [ {
```

```

"fd_name" : "nameCh",
"fd_value" : "测试8M数据标准导入111449",
"fd_id" : "1020622096960831488",
"directory_id" : "1082230138994827264",
"row_id" : "1082235079767502848",
"id" : "1082235079767502860",
"status" : "DRAFT",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2023-03-06T09:35:50+08:00",
"update_time" : "2024-04-11T21:05:51.366+08:00"
}, {
"fd_name" : "nameEn",
"fd_value" : "qn_test_1450_update",
"fd_id" : "1020622096985997312",
"directory_id" : "1082230138994827264",
"row_id" : "1082235079767502848",
"id" : "1082235079767502861",
"status" : "DRAFT",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2023-03-06T09:35:50+08:00",
"update_time" : "2024-04-11T21:05:51.366+08:00"
}, {
"fd_name" : "englishName",
"fd_value" : "",
"fd_id" : "1185628711836360704",
"directory_id" : "1082230138994827264",
"row_id" : "1082235079767502848",
"id" : "1228088651729272832",
"status" : "DRAFT",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-04-11T21:05:51.356+08:00",
"update_time" : "2024-04-11T21:05:51.356+08:00"
}, {
"fd_name" : "dataType",
"fd_value" : "STRING",
"fd_id" : "1020622097006968832",
"directory_id" : "1082230138994827264",
"row_id" : "1082235079767502848",
"id" : "1082235079767502862",
"status" : "DRAFT",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2023-03-06T09:35:50+08:00",
"update_time" : "2024-04-11T21:05:51.366+08:00"
}, {
"fd_name" : "dataLength",
"fd_value" : "100",
"fd_id" : "1020622097032134656",
"directory_id" : "1082230138994827264",
"row_id" : "1082235079767502848",
"id" : "1082235079767502849",
"status" : "DRAFT",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2023-03-06T09:35:50+08:00",
"update_time" : "2024-04-11T21:05:51.366+08:00"
}, {
"fd_name" : "hasAllowValueList",
"fd_value" : "false",
"fd_id" : "1020622097048911872",
"directory_id" : "1082230138994827264",
"row_id" : "1082235079767502848",
"id" : "1082235079767502850",
"status" : "DRAFT",
"create_by" : "test_uesr",
"update_by" : "test_uesr",

```

```
"create_time" : "2023-03-06T09:35:50+08:00",
"update_time" : "2024-04-11T21:05:51.366+08:00"
}, {
  "fd_name" : "allowList",
  "fd_value" : "",
  "fd_id" : "1020622097065689088",
  "directory_id" : "1082230138994827264",
  "row_id" : "1082235079767502848",
  "id" : "1082235079767502851",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2023-03-06T09:35:50+08:00",
  "update_time" : "2024-04-11T21:05:51.366+08:00"
}, {
  "fd_name" : "referCodeTable",
  "fd_value" : "",
  "fd_id" : "1020622097086660608",
  "directory_id" : "1082230138994827264",
  "row_id" : "1082235079767502848",
  "id" : "1082235079767502852",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2023-03-06T09:35:50+08:00",
  "update_time" : "2024-04-11T21:05:51.366+08:00"
}, {
  "fd_name" : "codeStandColumn",
  "fd_value" : null,
  "fd_id" : "1020622097103437824",
  "directory_id" : "1082230138994827264",
  "row_id" : "1082235079767502848",
  "id" : "1082235079767502853",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2023-03-06T09:35:50+08:00",
  "update_time" : "2023-03-06T09:35:50+08:00"
}, {
  "fd_name" : "dqcRule",
  "fd_value" : null,
  "fd_id" : "1020622097124409344",
  "directory_id" : "1082230138994827264",
  "row_id" : "1082235079767502848",
  "id" : "1228088651729272833",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-04-11T21:05:51.356+08:00",
  "update_time" : "2024-04-11T21:05:51.356+08:00"
}, {
  "fd_name" : "ruleOwner",
  "fd_value" : "zty_tess",
  "fd_id" : "1020622097141186560",
  "directory_id" : "1082230138994827264",
  "row_id" : "1082235079767502848",
  "id" : "1082235079767502854",
  "status" : "DRAFT",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2023-03-06T09:35:50+08:00",
  "update_time" : "2024-04-11T21:05:51.366+08:00"
}, {
  "fd_name" : "dataMonitorOwner",
  "fd_value" : "zty_tess",
  "fd_id" : "1020622097162158080",
  "directory_id" : "1082230138994827264",
  "row_id" : "1082235079767502848",
  "id" : "1082235079767502855",
```

```
"status": "DRAFT",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2023-03-06T09:35:50+08:00",
"update_time": "2024-04-11T21:05:51.366+08:00"
}, {
  "fd_name": "standardLevel",
  "fd_value": null,
  "fd_id": "1020622097178935296",
  "directory_id": "1082230138994827264",
  "row_id": "1082235079767502848",
  "id": "1082235079767502856",
  "status": "DRAFT",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": "2023-03-06T09:35:50+08:00",
  "update_time": "2023-03-06T09:35:50+08:00"
}, {
  "fd_name": "description",
  "fd_value": "测试8M数据标准导入的描述",
  "fd_id": "1020622097195712512",
  "directory_id": "1082230138994827264",
  "row_id": "1082235079767502848",
  "id": "1082235079767502857",
  "status": "DRAFT",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": "2023-03-06T09:35:50+08:00",
  "update_time": "2024-04-11T21:05:51.366+08:00"
}],
  "id": "1082235079767502848",
  "directory_id": "1082230138994827264",
  "directory_path": null,
  "row_id": null,
  "status": "DRAFT",
  "approval_info": null,
  "new_biz": null,
  "from_public": false,
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": "2023-03-06T09:35:50+08:00",
  "update_time": "2024-04-11T21:05:51.355+08:00"
}
}
```

### 状态码: 400

#### BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码: 401

#### Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码: 403

#### Forbidden

```
{
  "error_code": "DS.60xx",
```

```
"error_msg" : "The user does not have permission to call this API."  
}
```

**状态码：404**

Not Found

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

修改ID为1082235079767502848的数据标准信息。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class UpdateStandardSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateStandardRequest request = new UpdateStandardRequest();  
        request.withId("{id}");  
        StandElementValueVOList body = new StandElementValueVOList();  
        List<StandElementValueVO> listbodyValues = new ArrayList<>();  
        listbodyValues.add(  
            new StandElementValueVO()  
                .withFdName("nameCh")  
                .withFdValue("测试8M数据标准导入111449")  
                .withId("1082235079767502860")  
        );  
        listbodyValues.add(  
            new StandElementValueVO()
```

```
.withFdName("nameEn")
.withFdValue("qn_test_1450_update")
.withId("1082235079767502861")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("englishName")
        .withFdValue("")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("dataType")
        .withFdValue("STRING")
        .withId("1082235079767502862")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("dataLength")
        .withFdValue("100")
        .withId("1082235079767502849")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("hasAllowValueList")
        .withFdValue("false")
        .withId("1082235079767502850")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("allowList")
        .withFdValue("")
        .withId("1082235079767502851")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("referCodeTable")
        .withFdValue("")
        .withId("1082235079767502852")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("codeStandColumn")
        .withId("1082235079767502853")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("dqcRule")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("ruleOwner")
        .withFdValue("zty_tess")
        .withId("1082235079767502854")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("dataMonitorOwner")
        .withFdValue("zty_tess")
        .withId("1082235079767502855")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("standardLevel")
        .withId("1082235079767502856")
);
listbodyValues.add(
    new StandElementValueVO()
        .withFdName("description")
        .withFdValue("测试8M数据标准导入的描述")
);
```



```
        .withId("1082235079767502857")
    );
    body.withDirectoryId("1082230138994827264");
    body.withId("1082235079767502848");
    body.withValues(listbodyValues);
    request.withBody(body);
    try {
        UpdateStandardResponse response = client.updateStandard(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

修改ID为1082235079767502848的数据标准信息。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateStandardRequest()
        request.id = "{id}"
        listValuesbody = [
            StandElementValueVO(
                fd_name="nameCh",
                fd_value="测试8M数据标准导入111449",
                id="1082235079767502860"
            ),
            StandElementValueVO(
                fd_name="nameEn",
                fd_value="qn_test_1450_update",
                id="1082235079767502861"
            ),
            StandElementValueVO(
                fd_name="englishName",
                fd_value=""
            )
        ]
        request.setListValuesbody(listValuesbody)
    except exceptions.ApiErrorException as e:
        print(e.getHttpStatusCode())
        print(e.getRequestId())
        print(e.getErrorCode())
        print(e.getErrorMsg())
```

```
    ),
    StandElementValueVO(
        fd_name="dataType",
        fd_value="STRING",
        id="1082235079767502862"
    ),
    StandElementValueVO(
        fd_name="dataLength",
        fd_value="100",
        id="1082235079767502849"
    ),
    StandElementValueVO(
        fd_name="hasAllowValueList",
        fd_value="false",
        id="1082235079767502850"
    ),
    StandElementValueVO(
        fd_name="allowList",
        fd_value="",
        id="1082235079767502851"
    ),
    StandElementValueVO(
        fd_name="referCodeTable",
        fd_value="",
        id="1082235079767502852"
    ),
    StandElementValueVO(
        fd_name="codeStandColumn",
        id="1082235079767502853"
    ),
    StandElementValueVO(
        fd_name="dqcRule"
    ),
    StandElementValueVO(
        fd_name="ruleOwner",
        fd_value="zty_tess",
        id="1082235079767502854"
    ),
    StandElementValueVO(
        fd_name="dataMonitorOwner",
        fd_value="zty_tess",
        id="1082235079767502855"
    ),
    StandElementValueVO(
        fd_name="standardLevel",
        id="1082235079767502856"
    ),
    StandElementValueVO(
        fd_name="description",
        fd_value="测试8M数据标准导入的描述",
        id="1082235079767502857"
    )
]
request.body = StandElementValueVOList(
    directory_id="1082230138994827264",
    id="1082235079767502848",
    values=listValuesbody
)
response = client.update_standard(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

修改ID为1082235079767502848的数据标准信息。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateStandardRequest{}
    request.Id = "{id}"
    fdValueValues:= "测试8M数据标准导入111449"
    idValues:= "1082235079767502860"
    fdValueValues1:= "qn_test_1450_update"
    idValues1:= "1082235079767502861"
    fdValueValues2:= ""
    fdValueValues3:= "STRING"
    idValues2:= "1082235079767502862"
    fdValueValues4:= "100"
    idValues3:= "1082235079767502849"
    fdValueValues5:= "false"
    idValues4:= "1082235079767502850"
    fdValueValues6:= ""
    idValues5:= "1082235079767502851"
    fdValueValues7:= ""
    idValues6:= "1082235079767502852"
    idValues7:= "1082235079767502853"
    fdValueValues8:= "zty_tess"
    idValues8:= "1082235079767502854"
    fdValueValues9:= "zty_tess"
    idValues9:= "1082235079767502855"
    idValues10:= "1082235079767502856"
    fdValueValues10:= "测试8M数据标准导入的描述"
    idValues11:= "1082235079767502857"
    var listValuesbody = []model.StandElementValueVo{
        {
            FdName: "nameCh",
            FdValue: &fdValueValues,
            Id: &idValues,
        },
        {
            FdName: "nameEn",
            FdValue: &fdValueValues1,
            Id: &idValues1,
        },
    }
```

```
        FdName: "englishName",
        FdValue: &fdValueValues2,
    },
    {
        FdName: "dataType",
        FdValue: &fdValueValues3,
        Id: &idValues2,
    },
    {
        FdName: "dataLength",
        FdValue: &fdValueValues4,
        Id: &idValues3,
    },
    {
        FdName: "hasAllowValueList",
        FdValue: &fdValueValues5,
        Id: &idValues4,
    },
    {
        FdName: "allowList",
        FdValue: &fdValueValues6,
        Id: &idValues5,
    },
    {
        FdName: "referCodeTable",
        FdValue: &fdValueValues7,
        Id: &idValues6,
    },
    {
        FdName: "codeStandColumn",
        Id: &idValues7,
    },
    {
        FdName: "dqcRule",
    },
    {
        FdName: "ruleOwner",
        FdValue: &fdValueValues8,
        Id: &idValues8,
    },
    {
        FdName: "dataMonitorOwner",
        FdValue: &fdValueValues9,
        Id: &idValues9,
    },
    {
        FdName: "standardLevel",
        Id: &idValues10,
    },
    {
        FdName: "description",
        FdValue: &fdValueValues10,
        Id: &idValues11,
    },
}
idStandElementValueVoList:= "1082235079767502848"
request.Body = &model.StandElementValueVoList{
    DirectoryId: "1082230138994827264",
    Id: &idStandElementValueVoList,
    Values: listValuesbody,
}
response, err := client.UpdateStandard(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.3.5 查看数据标准详情

### 功能介绍

通过ID获取数据标准详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/standards/{id}

表 8-110 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

## 请求参数

表 8-111 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-112 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	数据标准查询返回data结果。

表 8-113 data

参数	参数类型	描述
value	<a href="#">StandElementValueVOList</a> object	属性值列表。

表 8-114 StandElementValueVOList

参数	参数类型	描述
values	Array of <b>StandElementValueVO</b> objects	属性信息。
id	String	数据标准的ID, ID字符串。
directory_id	String	标准所属目录, ID字符串。
directory_path	String	目录树。
row_id	String	标准行的ID, ID字符串。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理, 只读。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。

表 8-115 StandElementValueVO

参数	参数类型	描述
fd_name	String	属性名称。
fd_value	String	属性值。
fd_id	String	属性定义的ID, ID字符串。
directory_id	String	标准所属目录, ID字符串。
row_id	String	标准所属行, ID字符串。
id	String	数据标准的ID, ID字符串。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。

表 8-116 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。



参数	参数类型	描述
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-117 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-118 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-119 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-120 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-121 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询ID为1082235079767502848的数据标准详情。

GET https://{endpoint}/v2/{project\_id}/design/standards/1082235079767502848

## 响应示例

### 状态码：200

Success, 返回数据标准详情。

```
{
  "data": {
    "value": {
      "id": 0,
      "directory_id": "793889791589650432",
      "values": [ {
        "fd_name": "nameEn",
        "fd_value": "demo"
      }, {
        "fd_name": "dataType",
        "fd_value": "STRING"
      }, {
        "fd_name": "dataLength",
        "fd_value": "128"
      }, {
        "fd_name": "hasAllowValueList",
        "fd_value": false
      }, {
        "fd_name": "allowList",
        "fd_value": ""
      }, {
        "fd_name": "referCodeTable",
        "fd_value": "885123958788317184"
      }, {
        "fd_name": "codeStandColumn",
        "fd_value": "52470"
      }, {
        "fd_name": "dqcRule",
        "fd_value": "{}"
      }, {
        "fd_name": "ruleOwner",
        "fd_value": "liuxu"
      }, {
        "fd_name": "dataMonitorOwner",
        "fd_value": "liuxu"
      }, {
        "fd_name": "standardLevel",
        "fd_value": "domain"
      }, {
        "fd_name": "description",
        "fd_value": "这是一个demo"
      }
    ]
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
}
```



```
"error_msg" : "User authentication failed."
}
```

**状态码: 403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码: 404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowStandardByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowStandardByIdRequest request = new ShowStandardByIdRequest();
        request.withId("{id}");
        try {
            ShowStandardByIdResponse response = client.showStandardById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowStandardByIdRequest()
        request.id = "{id}"
        response = client.show_standard_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```

projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowStandardByIdRequest{
    request.Id = "{id}"
}
response, err := client.ShowStandardById(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回数据标准详情。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.3.6 关联属性与数据标准

### 功能介绍

关联属性与数据标准。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/standards/attribute

表 8-122 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-123 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-124 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	属性ID列表，ID字符串。
stand_row_id	是	String	关联的数据标准的ID，ID字符串。
table_id	是	String	表ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	是	String	表类型，默认是 TABLE_MODEL。 枚举值： <ul style="list-style-type: none"> <li>TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>FACT_LOGIC_TABLE: 事实表</li> <li>DIMENSION: 维度</li> <li>DIMENSION_LOGIC_TABLE: 维度表</li> </ul>

## 响应参数

状态码：200

表 8-125 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-126 data

参数	参数类型	描述
value	<b>LinkAttributeAndElementVO</b> object	属性ID列表。

表 8-127 LinkAttributeAndElementVO

参数	参数类型	描述
ids	Array of strings	属性ID列表，ID字符串。
stand_row_id	String	关联的数据标准的ID，ID字符串。
table_id	String	表ID，ID字符串。

参数	参数类型	描述
biz_type	String	表类型，默认是TABLE_MODEL。 枚举值： <ul style="list-style-type: none"> <li>TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>FACT_LOGIC_TABLE: 事实表</li> <li>DIMENSION: 维度</li> <li>DIMENSION_LOGIC_TABLE: 维度表</li> </ul>

状态码：400

表 8-128 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-129 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-130 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

**状态码：404**

表 8-131 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，将表的属性与数据标准关联。

PUT https://{endpoint}/v2/{project\_id}/design/standards/attribute

```
{
  "ids" : [ "1229920251374923780" ],
  "stand_row_id" : "1169318364674498561",
  "table_id" : "1229920251379118081",
  "biz_type" : "TABLE_MODEL"
}
```

## 响应示例

**状态码：200**

Success

```
{
  "data" : {
    "value" : {
      "ids" : [ "1229920251374923780" ],
      "stand_row_id" : "1169318364674498561",
      "table_id" : "1229920251379118081",
      "biz_type" : "TABLE_MODEL"
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，将表的属性与数据标准关联。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ResetLinkAttributeAndStandardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetLinkAttributeAndStandardRequest request = new ResetLinkAttributeAndStandardRequest();
```



```
LinkAttributeAndElementVO body = new LinkAttributeAndElementVO();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1229920251374923780");
body.withBizType(LinkAttributeAndElementVO.BizTypeEnum.fromValue("TABLE_MODEL"));
body.withTableId("1229920251379118081");
body.withStandRowId("1169318364674498561");
body.withIds(listbodyIds);
request.withBody(body);
try {
    ResetLinkAttributeAndStandardResponse response = client.resetLinkAttributeAndStandard(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据请求参数，将表的属性与数据标准关联。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetLinkAttributeAndStandardRequest()
        listbody = [
            "1229920251374923780"
        ]
        request.body = LinkAttributeAndElementVO(
            biz_type="TABLE_MODEL",
            table_id="1229920251379118081",
            stand_row_id="1169318364674498561",
            ids=listbody
        )
        response = client.reset_link_attribute_and_standard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

根据请求参数，将表的属性与数据标准关联。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetLinkAttributeAndStandardRequest{}
    var listIdsbody = []string{
        "1229920251374923780",
    }
    request.Body = &model.LinkAttributeAndElementVo{
        BizType: model.GetLinkAttributeAndElementVoBizTypeEnum().TABLE_MODEL,
        TableId: "1229920251379118081",
        StandRowId: "1169318364674498561",
        Ids: listIdsbody,
    }
    response, err := client.ResetLinkAttributeAndStandard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.4 数据源接口

### 8.4.1 获取数据连接信息

#### 功能介绍

获取指定类型下的数据连接信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/atlas/data-warehouses

表 8-132 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-133 Query 参数

参数	是否必选	参数类型	描述
force_refresh	否	Boolean	是否查询最新的。
dw_type	是	String	数据连接类型。
limit	否	Integer	limit
offset	否	Integer	limit

## 请求参数

表 8-134 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-135 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-136 data

参数	参数类型	描述
value	Array of <a href="#">DataConnection VO</a> objects	数据连接信息数组

表 8-137 DataConnectionVO

参数	参数类型	描述
dw_name	String	数据连接名称。
dw_id	String	数据连接ID。
display_name	String	数据连接名称，适配现有实现。
dw_type	String	数据连接类型。

状态码：400

表 8-138 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-139 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-140 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

获取数据连接类型为DWS的数据连接信息。

```
GET https://{endpoint}/v2/{project_id}/design/atlas/data-warehouses?dw_type=DWS
```

## 响应示例

**状态码：200**

Success, 返回data数据是指定类型下所有数据连接DataConnectionVO实体。

```
{
  "data": {
    "value": [ {
      "dw_id": "c5daea963128457cb7579404da5a23c7",
      "dw_name": "dws_test",
      "dw_type": "DWS"
    } ]
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class SearchDwByTypeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        SearchDwByTypeRequest request = new SearchDwByTypeRequest();
        try {
            SearchDwByTypeResponse response = client.searchDwByType(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchDwByTypeRequest()
```

```

response = client.search_dw_by_type(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchDwByTypeRequest{}
    response, err := client.SearchDwByType(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是指定类型下所有数据连接DataConnectionVO实体。
400	BadRequest



状态码	描述
401	Unauthorized
403	Forbidden

## 8.5 码表管理接口

### 8.5.1 查询码表列表

#### 功能介绍

查询码表列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/code-tables

表 8-141 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-142 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
name_ch	否	String	按名称精确查询。
name_en	否	String	按编码精确查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。
directory_id	否	String	目录ID，ID字符串。

参数	是否必选	参数类型	描述
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-143 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-144 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-145 data

参数	参数类型	描述
value	<b>value</b> object	value，统一的返回结果的外层数据结构。

表 8-146 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>CodeTableVO</b> objects	查询到的码表对象（CodeTableVO）数组。

表 8-147 CodeTableVO

参数	参数类型	描述
id	String	码表ID, ID字符串。
name_en	String	表名称, 英文名。
name_ch	String	表名称, 中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-148 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-149 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-150 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-151 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

状态码: 400

表 8-152 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-153 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-154 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-155 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

获取前十条码表列表信息。

GET https://{endpoint}/v2/{project\_id}/design/code-tables?offset=1&limit=10

## 响应示例

### 状态码：200

Success, 返回data数据是CodeTableVO 数组, 以及总数。

```
{
  "data": {
    "value": {
      "total": 10,
      "records": [ {
        "id": "1014131824240230400",
        "name_en": "test",
        "name_ch": "test",
        "tb_version": 0,
        "directory_id": "1012307270173851648",
        "directory_path": null,
        "description": "",
        "create_by": "abc",
        "status": "DRAFT",
        "create_time": "2022-08-30T11:17:48.557+08:00",
        "update_time": "2022-08-30T11:17:48.557+08:00",
        "approval_info": null,
        "new_biz": null,
        "code_table_fields": [ {
          "id": null,
          "code_table_id": null,
          "ordinal": 1,
          "name_en": "code",
          "name_ch": "编码",
          "description": "",
          "data_type": "STRING",
          "domain_type": null,
          "data_type_extend": null,
          "is_unique_key": false,
          "code_table_field_values": [ ],
          "count_field_values": null
        }, {
          "id": null,
          "code_table_id": null,
          "ordinal": 2,
          "name_en": "value",
          "name_ch": "值",
          "description": "",
          "data_type": "STRING",
          "domain_type": null,
          "data_type_extend": null,
          "is_unique_key": false,
          "code_table_field_values": [ ],
          "count_field_values": null
        }
      ]
    }
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

### 状态码：404

### Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchCodeTablesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        SearchCodeTablesRequest request = new SearchCodeTablesRequest();
        try {
            SearchCodeTablesResponse response = client.searchCodeTables(request);
            System.out.println(response.toString());
        }
    }
}
```

```
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchCodeTablesRequest()
        response = client.search_code_tables(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```

ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.SearchCodeTablesRequest{}
response, err := client.SearchCodeTables(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是CodeTableVO 数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.5.2 创建码表

### 功能介绍

创建码表。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/code-tables



表 8-156 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-157 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-158 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	码表ID，ID字符串。
name_en	是	String	表名称，英文名。
name_ch	是	String	表名称，中文名。
tb_version	否	Integer	表版本。
directory_id	是	String	目录ID，ID字符串。
directory_path	否	String	目录树。
description	否	String	描述。

参数	是否必选	参数类型	描述
create_by	否	String	创建人。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	否	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	<b>BizVersionMa</b> <b>nageVO</b> object	业务版本管理，只读。
code_table_fields	是	Array of <b>CodeTableFie</b> <b>ldVO</b> objects	码表属性信息。

表 8-159 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-160 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。



参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-161 CodeTableFieldVO

参数	是否必选	参数类型	描述
id	否	String	码表字段ID，ID字符串。
code_table_id	否	String	所属码表ID（更新时必须填），ID字符串。
ordinal	是	Integer	序号。
name_en	是	String	字段名，英文。
name_ch	是	String	字段名，中文。
description	否	String	描述。
data_type	是	String	字段类型。
data_type_ext_end	否	String	数据类型扩展字段。
is_unique_key	否	Boolean	是否唯一。
code_table_field_values	否	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	否	Integer	码表属性值总数。

表 8-162 CodeTableFieldValueVO

参数	是否必选	参数类型	描述
id	否	String	码表字段ID, ID字符串。
fd_id	否	String	所属码表属性ID, ID字符串。
fd_value	否	String	码表属性值。
ordinal	否	Integer	序号。
description	否	String	描述。

## 响应参数

状态码: 200

表 8-163 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-164 data

参数	参数类型	描述
value	<b>CodeTableVO</b> object	码表结构。

表 8-165 CodeTableVO

参数	参数类型	描述
id	String	码表ID, ID字符串。
name_en	String	表名称, 英文名。
name_ch	String	表名称, 中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-166 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-167 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-168 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-169 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

状态码: 400

表 8-170 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-171 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-172 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-173 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，创建一个新的码表，保存到草稿，返回信息是创建成功的码表信息。

POST https://{endpoint}/v2/{project\_id}/design/code-tables

```
{
  "name_en": "",
  "name_ch": "test_create_code_table",
  "description": "",
  "code_table_fields": [ {
    "ordinal": 1,
    "name_ch": "编码",
    "name_en": "code",
    "data_type": "STRING",
    "description": "",
    "code_table_field_values": [ ]
  }, {
    "ordinal": 2,
    "name_ch": "省",
    "name_en": "province",
    "data_type": "STRING",
    "description": "",
    "code_table_field_values": [ ]
  }, {
    "ordinal": 3,
    "name_ch": "市",
    "name_en": "city",
    "data_type": "STRING",
    "description": "",
    "code_table_field_values": [ ]
  }, {
    "ordinal": 4,
    "name_ch": "县",
    "name_en": "county",
    "data_type": "STRING",
    "description": "",
    "code_table_field_values": [ ]
  } ],
  "tb_version": 0,
  "directory_id": "1194963842254491648"
}
```

## 响应示例

**状态码：200**

Success, 返回data数据是CodeTableVO。

```
{
  "data": {
    "value": {
      "id": "1230204979835502592",
      "name_en": "DC000007",
      "name_ch": "test_create_code_table",
      "tb_version": 0,
      "directory_id": "1194963842254491648",
      "directory_path": null,
      "description": "",
      "create_by": "test_uesr",
      "status": "DRAFT",
      "create_time": "2024-04-17T17:15:23.296+08:00",
      "update_time": "2024-04-17T17:15:23.296+08:00",
      "approval_info": null,
      "new_biz": null,
      "code_table_fields": [ {
        "id": null,
        "code_table_id": null,
        "ordinal": 1,
        "name_en": "code",
        "name_ch": "编码",
        "description": "",
        "data_type": "STRING",
```

```
"domain_type" : null,
"data_type_extend" : null,
"is_unique_key" : false,
"code_table_field_values" : [ ],
"count_field_values" : null
}, {
  "id" : null,
  "code_table_id" : null,
  "ordinal" : 2,
  "name_en" : "province",
  "name_ch" : "省",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}, {
  "id" : null,
  "code_table_id" : null,
  "ordinal" : 3,
  "name_en" : "city",
  "name_ch" : "市",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}, {
  "id" : null,
  "code_table_id" : null,
  "ordinal" : 4,
  "name_en" : "county",
  "name_ch" : "县",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}
}
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，创建一个新的码表，保存到草稿，返回信息是创建成功的码表信息。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateCodeTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateCodeTableRequest request = new CreateCodeTableRequest();
        CodeTableVO body = new CodeTableVO();
        List<CodeTableFieldVO> listbodyCodeTableFields = new ArrayList<>();
        listbodyCodeTableFields.add(
            new CodeTableFieldVO()
                .withOrdinal(1)
                .withNameEn("code")
                .withNameCh("编码")
                .withDescription("")
                .withDataType("STRING")
        );
    }
}
```

```
        .withCodeTableFieldValues()
    );
    listbodyCodeTableFields.add(
        new CodeTableFieldVO()
            .withOrdinal(2)
            .withNameEn("province")
            .withNameCh("省")
            .withDescription("")
            .withDataType("STRING")
            .withCodeTableFieldValues()
    );
    listbodyCodeTableFields.add(
        new CodeTableFieldVO()
            .withOrdinal(3)
            .withNameEn("city")
            .withNameCh("市")
            .withDescription("")
            .withDataType("STRING")
            .withCodeTableFieldValues()
    );
    listbodyCodeTableFields.add(
        new CodeTableFieldVO()
            .withOrdinal(4)
            .withNameEn("county")
            .withNameCh("县")
            .withDescription("")
            .withDataType("STRING")
            .withCodeTableFieldValues()
    );
    body.withCodeTableFields(listbodyCodeTableFields);
    body.withDescription("");
    body.withDirectoryId("1194963842254491648");
    body.withTbVersion(0);
    body.withNameCh("test_create_code_table");
    body.withNameEn("");
    request.withBody(body);
    try {
        CreateCodeTableResponse response = client.createCodeTable(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
    }
}
```

## Python

根据请求参数，创建一个新的码表，保存到草稿，返回信息是创建成功的码表信息。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateCodeTableRequest()
    listCodeTableFieldsbody = [
        CodeTableFieldVO(
            ordinal=1,
            name_en="code",
            name_ch="编码",
            description="",
            data_type="STRING",
        ),
        CodeTableFieldVO(
            ordinal=2,
            name_en="province",
            name_ch="省",
            description="",
            data_type="STRING",
        ),
        CodeTableFieldVO(
            ordinal=3,
            name_en="city",
            name_ch="市",
            description="",
            data_type="STRING",
        ),
        CodeTableFieldVO(
            ordinal=4,
            name_en="county",
            name_ch="县",
            description="",
            data_type="STRING",
        )
    ]
    request.body = CodeTableVO(
        code_table_fields=listCodeTableFieldsbody,
        description="",
        directory_id="1194963842254491648",
        tb_version=0,
        name_ch="test_create_code_table",
        name_en=""
    )
    response = client.create_code_table(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据请求参数，创建一个新的码表，保存到草稿，返回信息是创建成功的码表信息。

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateCodeTableRequest{}
    descriptionCodeTableFields:= ""
    descriptionCodeTableFields1:= ""
    descriptionCodeTableFields2:= ""
    descriptionCodeTableFields3:= ""
    var listCodeTableFieldsbody = []model.CodeTableFieldVo{
        {
            Ordinal: int32(1),
            NameEn: "code",
            NameCh: "编码",
            Description: &descriptionCodeTableFields,
            DataType: "STRING",
        },
        {
            Ordinal: int32(2),
            NameEn: "province",
            NameCh: "省",
            Description: &descriptionCodeTableFields1,
            DataType: "STRING",
        },
        {
            Ordinal: int32(3),
            NameEn: "city",
            NameCh: "市",
            Description: &descriptionCodeTableFields2,
            DataType: "STRING",
        },
        {
            Ordinal: int32(4),
            NameEn: "county",
            NameCh: "县",
            Description: &descriptionCodeTableFields3,
            DataType: "STRING",
        },
    }
    descriptionCodeTableVo:= ""
    tbVersionCodeTableVo:= int32(0)
    request.Body = &model.CodeTableVo{
        CodeTableFields: listCodeTableFieldsbody,
        Description: &descriptionCodeTableVo,
        DirectoryId: "1194963842254491648",
    }
}
```



```
TbVersion: &tbVersionCodeTableVo,
NameCh: "test_create_code_table",
NameEn: "",
}
response, err := client.CreateCodeTable(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是CodeTableVO。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.5.3 删除码表

### 功能介绍

删除码表。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/code-tables

表 8-174 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-175 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-176 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-177 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-178 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-179 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-180 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-181 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

表 8-182 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**请求示例**

根据请求参数，删除码表，返回的是成功删除的码表条数。

```
DELETE https://{endpoint}/v2/{project_id}/design/code-tables
{
  "ids" : [ "1230206616507449344", "1230206580511875072" ]
}
```

**响应示例**

**状态码：200**

Success，返回data数据是删除个数。

```
{
  "value" : 2
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

## Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，删除码表，返回的是成功删除的码表条数。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteCodeTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteCodeTableRequest request = new DeleteCodeTableRequest();
        IdsParam body = new IdsParam();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("1230206616507449344");
        listbodyIds.add("1230206580511875072");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            DeleteCodeTableResponse response = client.deleteCodeTable(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
        }
    }
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

根据请求参数，删除码表，返回的是成功删除的码表条数。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteCodeTableRequest()
        listIdsbody = [
            "1230206616507449344",
            "1230206580511875072"
        ]
        request.body = IdsParam(
            ids=listIdsbody
        )
        response = client.delete_code_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据请求参数，删除码表，返回的是成功删除的码表条数。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteCodeTableRequest{}
var listIdsbody = []string{
    "1230206616507449344",
    "1230206580511875072",
}
request.Body = &model.IdsParam{
    Ids: listIdsbody,
}
response, err := client.DeleteCodeTable(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.5.4 修改码表

### 功能介绍

修改码表。

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v2/{project\_id}/design/code-tables/{id}

表 8-183 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

## 请求参数

表 8-184 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-185 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	码表ID，ID字符串。
name_en	是	String	表名称，英文名。



参数	是否必选	参数类型	描述
name_ch	是	String	表名称, 中文名。
tb_version	否	Integer	表版本。
directory_id	是	String	目录ID, ID字符串。
directory_path	否	String	目录树。
description	否	String	描述。
create_by	否	String	创建人。
status	否	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	否	<b>ApprovalVO</b> object	审批信息, 只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	<b>BizVersionMa</b> <b>nageVO</b> object	业务版本管理, 只读。
code_table_fields	是	Array of <b>CodeTableFie</b> <b>ldVO</b> objects	码表属性信息。

表 8-186 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID, ID字符串。
tenant_id	否	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。

参数	是否必选	参数类型	描述
biz_id	否	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-187 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。



参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-188 CodeTableFieldVO

参数	是否必选	参数类型	描述
id	否	String	码表字段ID，ID字符串。
code_table_id	否	String	所属码表ID（更新时必须填），ID字符串。
ordinal	是	Integer	序号。
name_en	是	String	字段名，英文。
name_ch	是	String	字段名，中文。
description	否	String	描述。
data_type	是	String	字段类型。
data_type_ext_end	否	String	数据类型扩展字段。
is_unique_key	否	Boolean	是否唯一。
code_table_field_values	否	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	否	Integer	码表属性值总数。

表 8-189 CodeTableFieldValueVO

参数	是否必选	参数类型	描述
id	否	String	码表字段ID, ID字符串。
fd_id	否	String	所属码表属性ID, ID字符串。
fd_value	否	String	码表属性值。
ordinal	否	Integer	序号。
description	否	String	描述。

## 响应参数

状态码: 200

表 8-190 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-191 data

参数	参数类型	描述
value	<b>CodeTableVO</b> object	码表结构。

表 8-192 CodeTableVO

参数	参数类型	描述
id	String	码表ID, ID字符串。
name_en	String	表名称, 英文名。
name_ch	String	表名称, 中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-193 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-194 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-195 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-196 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

状态码: 400

表 8-197 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-198 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-199 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-200 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，修改码表，返回的是修改后的码表信息。

```
PUT https://{endpoint}/v2/{project_id}/design/code-tables/1230204979835502592
```

```
{
  "name_en": "DC000007",
  "name_ch": "test_update_code_table",
  "description": "",
  "code_table_fields": [ {
    "ordinal": 1,
    "name_ch": "编码",
    "name_en": "code",
    "data_type": "STRING",
    "description": "",
    "code_table_field_values": [ ],
    "code_table_id": "1230204979835502592",
    "id": "178101"
  }, {
    "ordinal": 2,
    "name_ch": "省",
    "name_en": "province",
    "data_type": "STRING",
    "description": "",
    "code_table_field_values": [ ],
    "code_table_id": "1230204979835502592",
    "id": "178102"
  }, {
    "ordinal": 3,
    "name_ch": "市",
    "name_en": "city",
    "data_type": "STRING",
    "description": "",
    "code_table_field_values": [ ],
    "code_table_id": "1230204979835502592",
    "id": "178103"
  }, {
    "ordinal": 4,
    "name_ch": "县",
    "name_en": "county",
    "data_type": "STRING",
    "description": "",
    "code_table_field_values": [ ],
    "code_table_id": "1230204979835502592",
    "id": "178104"
  } ],
  "tb_version": 0,
  "directory_id": "1194963842254491648",
  "id": "1230204979835502592"
}
```

## 响应示例

**状态码: 200**

Success, 返回data数据是CodeTableVO。

```
{
  "data": {
    "value": {
      "id": "1230204979835502592",
      "name_en": "DC000007",
      "name_ch": "test_update_code_table",
      "tb_version": 0,
      "directory_id": "1194963842254491648",
      "directory_path": null,
      "description": "",
      "create_by": "test_uesr",
      "status": "DRAFT",
      "create_time": "2024-04-17T17:15:23+08:00",
      "update_time": "2024-04-17T17:25:34.449+08:00",
      "approval_info": null,
    }
  }
}
```

```

"new_biz" : null,
"code_table_fields" : [ {
  "id" : "178101",
  "code_table_id" : "1230204979835502592",
  "ordinal" : 1,
  "name_en" : "code",
  "name_ch" : "编码",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}, {
  "id" : "178102",
  "code_table_id" : "1230204979835502592",
  "ordinal" : 2,
  "name_en" : "province",
  "name_ch" : "省",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}, {
  "id" : "178103",
  "code_table_id" : "1230204979835502592",
  "ordinal" : 3,
  "name_en" : "city",
  "name_ch" : "市",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}, {
  "id" : "178104",
  "code_table_id" : "1230204979835502592",
  "ordinal" : 4,
  "name_en" : "county",
  "name_ch" : "县",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}
]
}
}
}

```

**状态码: 400**

BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

**状态码: 401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，修改码表，返回的是修改后的码表信息。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateCodeTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateCodeTableRequest request = new UpdateCodeTableRequest();
```

```
request.withId("{id}");
CodeTableVO body = new CodeTableVO();
List<CodeTableFieldVO> listbodyCodeTableFields = new ArrayList<>();
listbodyCodeTableFields.add(
    new CodeTableFieldVO()
        .withId("178101")
        .withCodeTableId("1230204979835502592")
        .withOrdinal(1)
        .withNameEn("code")
        .withNameCh("编码")
        .withDescription("")
        .withDataType("STRING")
        .withCodeTableFieldValues()
);
listbodyCodeTableFields.add(
    new CodeTableFieldVO()
        .withId("178102")
        .withCodeTableId("1230204979835502592")
        .withOrdinal(2)
        .withNameEn("province")
        .withNameCh("省")
        .withDescription("")
        .withDataType("STRING")
        .withCodeTableFieldValues()
);
listbodyCodeTableFields.add(
    new CodeTableFieldVO()
        .withId("178103")
        .withCodeTableId("1230204979835502592")
        .withOrdinal(3)
        .withNameEn("city")
        .withNameCh("市")
        .withDescription("")
        .withDataType("STRING")
        .withCodeTableFieldValues()
);
listbodyCodeTableFields.add(
    new CodeTableFieldVO()
        .withId("178104")
        .withCodeTableId("1230204979835502592")
        .withOrdinal(4)
        .withNameEn("county")
        .withNameCh("县")
        .withDescription("")
        .withDataType("STRING")
        .withCodeTableFieldValues()
);
body.withCodeTableFields(listbodyCodeTableFields);
body.withDescription("");
body.withDirectoryId("1194963842254491648");
body.withTbVersion(0);
body.withNameCh("test_update_code_table");
body.withNameEn("DC000007");
body.withId("1230204979835502592");
request.withBody(body);
try {
    UpdateCodeTableResponse response = client.updateCodeTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

```
}  
}
```

## Python

根据请求参数，修改码表，返回的是修改后的码表信息。

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UpdateCodeTableRequest()  
        request.id = "{id}"  
        listCodeTableFieldsbody = [  
            CodeTableFieldVO(  
                id="178101",  
                code_table_id="1230204979835502592",  
                ordinal=1,  
                name_en="code",  
                name_ch="编码",  
                description="",  
                data_type="STRING",  
            ),  
            CodeTableFieldVO(  
                id="178102",  
                code_table_id="1230204979835502592",  
                ordinal=2,  
                name_en="province",  
                name_ch="省",  
                description="",  
                data_type="STRING",  
            ),  
            CodeTableFieldVO(  
                id="178103",  
                code_table_id="1230204979835502592",  
                ordinal=3,  
                name_en="city",  
                name_ch="市",  
                description="",  
                data_type="STRING",  
            ),  
            CodeTableFieldVO(  
                id="178104",  
                code_table_id="1230204979835502592",  
                ordinal=4,  
                name_en="county",  
                name_ch="县",
```

```
        description="",
        data_type="STRING",
    )
]
request.body = CodeTableVO(
    code_table_fields=listCodeTableFieldsbody,
    description="",
    directory_id="1194963842254491648",
    tb_version=0,
    name_ch="test_update_code_table",
    name_en="DC000007",
    id="1230204979835502592"
)
response = client.update_code_table(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据请求参数，修改码表，返回的是修改后的码表信息。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateCodeTableRequest{}
    request.Id = "{id}"
    idCodeTableFields:= "178101"
    codeTableIdCodeTableFields:= "1230204979835502592"
    descriptionCodeTableFields:= ""
    idCodeTableFields1:= "178102"
    codeTableIdCodeTableFields1:= "1230204979835502592"
    descriptionCodeTableFields1:= ""
    idCodeTableFields2:= "178103"
    codeTableIdCodeTableFields2:= "1230204979835502592"
    descriptionCodeTableFields2:= ""
    idCodeTableFields3:= "178104"
```



```
codeTableIdCodeTableFields3:= "1230204979835502592"
descriptionCodeTableFields3:= ""
var listCodeTableFieldsbody = []model.CodeTableFieldVo{
    {
        Id: &idCodeTableFields,
        CodeTableId: &codeTableIdCodeTableFields,
        Ordinal: int32(1),
        NameEn: "code",
        NameCh: "编码",
        Description: &descriptionCodeTableFields,
        DataType: "STRING",
    },
    {
        Id: &idCodeTableFields1,
        CodeTableId: &codeTableIdCodeTableFields1,
        Ordinal: int32(2),
        NameEn: "province",
        NameCh: "省",
        Description: &descriptionCodeTableFields1,
        DataType: "STRING",
    },
    {
        Id: &idCodeTableFields2,
        CodeTableId: &codeTableIdCodeTableFields2,
        Ordinal: int32(3),
        NameEn: "city",
        NameCh: "市",
        Description: &descriptionCodeTableFields2,
        DataType: "STRING",
    },
    {
        Id: &idCodeTableFields3,
        CodeTableId: &codeTableIdCodeTableFields3,
        Ordinal: int32(4),
        NameEn: "county",
        NameCh: "县",
        Description: &descriptionCodeTableFields3,
        DataType: "STRING",
    },
}
descriptionCodeTableVo:= ""
tbVersionCodeTableVo:= int32(0)
idCodeTableVo:= "1230204979835502592"
request.Body = &model.CodeTableVo{
    CodeTableFields: listCodeTableFieldsbody,
    Description: &descriptionCodeTableVo,
    DirectoryId: "1194963842254491648",
    TbVersion: &tbVersionCodeTableVo,
    NameCh: "test_update_code_table",
    NameEn: "DC000007",
    Id: &idCodeTableVo,
}
response, err := client.UpdateCodeTable(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是CodeTableVO。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.5.5 查看码表详情

### 功能介绍

通过ID查看码表的详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/code-tables/{id}

表 8-201 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID, ID字符串。

### 请求参数

表 8-202 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 ( 响应消息头中X-Subject-Token的值) 使用Token认证时必选。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-203 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-204 data

参数	参数类型	描述
value	<a href="#">CodeTableVO</a> object	码表结构。

表 8-205 CodeTableVO

参数	参数类型	描述
id	String	码表ID，ID字符串。
name_en	String	表名称，英文名。
name_ch	String	表名称，中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID，ID字符串。
directory_path	String	目录树。

参数	参数类型	描述
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-206 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-207 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-208 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-209 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

状态码: 400

表 8-210 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-211 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-212 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-213 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据码表ID，查询码表信息。

GET https://{endpoint}/v2/{project\_id}/design/code-tables/1230204979835502592

## 响应示例

### 状态码：200

Success, 返回data数据是CodeTableVO详情。

```
{
  "data": {
    "value": {
      "id": "1012307352952635392",
      "name_en": "RY_000001",
      "name_ch": "性别",
      "tb_version": 0,
      "directory_id": "1012307270173851648",
      "directory_path": null,
      "description": "",
      "create_by": "abc",
      "status": "PUBLISHED",
      "create_time": "2022-08-25T10:28:01+08:00",
      "update_time": "2022-08-25T10:31:08+08:00",
      "approval_info": null,
      "new_biz": null,
      "code_table_fields": [ {
        "id": "66929",
        "code_table_id": "1012307352952635392",
        "ordinal": 1,
        "name_en": "code",
        "name_ch": "编码",
        "description": "",
        "data_type": "STRING",
        "domain_type": null,
        "data_type_extend": null,
        "is_unique_key": false,
        "code_table_field_values": [ ],
        "count_field_values": null
      }, {
        "id": "66930",
        "code_table_id": "1012307352952635392",
        "ordinal": 2,
        "name_en": "value",
        "name_ch": "值",
        "description": "",
        "data_type": "STRING",
        "domain_type": null,
        "data_type_extend": null,
        "is_unique_key": false,
        "code_table_field_values": [ ],
        "count_field_values": null
      } ]
    }
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowCodeTableByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowCodeTableByIdRequest request = new ShowCodeTableByIdRequest();
        request.withId("{id}");
        try {
            ShowCodeTableByIdResponse response = client.showCodeTableById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCodeTableByIdRequest()
        request.id = "{id}"
        response = client.show_code_table_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```

ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowCodeTableByIdRequest{}
request.Id = "{id}"
response, err := client.ShowCodeTableById(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是CodeTableVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.5.6 查看码表字段值

### 功能介绍

查看码表字段值。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/code-tables/{id}/values

表 8-214 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-215 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-216 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。



## 响应参数

**状态码：200**

表 8-217 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

**状态码：400**

表 8-218 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-219 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-220 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

表 8-221 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，编辑码表的字段值，包括码表字段的新增。

GET https://{endpoint}/v2/{project\_id}/design/code-tables/1230204979835502592/values

## 响应示例

**状态码：200**

Success，返回data数据是CodeTableFieldInfoVO数组，以及总数。

```
{
  "data": {
    "value": {
      "id": "1230204979835502592",
      "name_en": "DC000007",
      "name_ch": "test_update_code_table",
      "tb_version": 0,
      "directory_id": "1194963842254491648",
      "directory_path": "test_czh_dir",
      "description": "",
      "from_public": false,
      "create_by": "test_uesr",
      "status": "DRAFT",
      "create_time": "2024-04-17T17:15:23+08:00",
      "update_time": "2024-04-17T17:25:34+08:00",
      "approval_info": null,
      "new_biz": null,
      "code_table_fields": [ {
        "id": "178101",
        "code_table_id": "1230204979835502592",
        "ordinal": 1,
        "name_en": "code",
        "name_ch": "编码",
        "description": "",
        "data_type": "STRING",
        "domain_type": null,
        "data_type_extend": null,
        "is_unique_key": false,
        "code_table_field_values": [ ],
        "count_field_values": null
      }, {
        "id": "178102",
        "code_table_id": "1230204979835502592",
        "ordinal": 2,
        "name_en": "province",
        "name_ch": "省",
        "description": "",
        "data_type": "STRING",
        "domain_type": null,
        "data_type_extend": null,
        "is_unique_key": false,
        "code_table_field_values": [ ],

```

```
"count_field_values" : null
}, {
  "id" : "178103",
  "code_table_id" : "1230204979835502592",
  "ordinal" : 3,
  "name_en" : "city",
  "name_ch" : "市",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}, {
  "id" : "178104",
  "code_table_id" : "1230204979835502592",
  "ordinal" : 4,
  "name_en" : "county",
  "name_ch" : "县",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ ],
  "count_field_values" : null
}]
}
}
```

#### 状态码：400

##### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

##### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

##### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

#### 状态码：404

##### Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchCodeTableValuesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        SearchCodeTableValuesRequest request = new SearchCodeTableValuesRequest();
        request.withId("{id}");
        try {
            SearchCodeTableValuesResponse response = client.searchCodeTableValues(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchCodeTableValuesRequest()
        request.id = "{id}"
        response = client.search_code_table_values(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchCodeTableValuesRequest{}
    request.Id = "{id}"
    response, err := client.SearchCodeTableValues(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是CodeTableFieldInfoVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.5.7 编辑码表字段值

### 功能介绍

编辑码表字段值。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/code-tables/{id}/values

表 8-222 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

## 请求参数

表 8-223 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-224 请求 Body 参数

参数	是否必选	参数类型	描述
to_add	否	Array of <a href="#">CodeTableFieldVO</a> objects	新增码表属性、属性值列表。
to_modify	否	Array of <a href="#">CodeTableFieldVO</a> objects	编辑码表属性值列表。
to_remove	否	Array of <a href="#">CodeTableFieldVO</a> objects	删除码表属性ID列表。

表 8-225 CodeTableFieldVO

参数	是否必选	参数类型	描述
id	否	String	码表字段ID，ID字符串。
code_table_id	否	String	所属码表ID（更新时必填），ID字符串。

参数	是否必选	参数类型	描述
ordinal	是	Integer	序号。
name_en	是	String	字段名，英文。
name_ch	是	String	字段名，中文。
description	否	String	描述。
data_type	是	String	字段类型。
data_type_ext end	否	String	数据类型扩展字段。
is_unique_key	否	Boolean	是否唯一。
code_table_fie ld_values	否	Array of <a href="#">CodeTableFie ldValueVO</a> objects	码表属性值。
count_field_va lues	否	Integer	码表属性值总数。

表 8-226 CodeTableFieldValueVO

参数	是否必选	参数类型	描述
id	否	String	码表字段ID，ID字符串。
fd_id	否	String	所属码表属性ID，ID字符串。
fd_value	否	String	码表属性值。
ordinal	否	Integer	序号。
description	否	String	描述。

## 响应参数

状态码：200

表 8-227 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。



表 8-228 data

参数	参数类型	描述
value	Array of <a href="#">CodeTableFieldVO</a> objects	码表字段列表信息。

表 8-229 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-230 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。

参数	参数类型	描述
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

**状态码：400**

表 8-231 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-232 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-233 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

表 8-234 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，编辑码表的字段值，包括码表字段的新增。

PUT https://{endpoint}/v2/{project\_id}/design/code-tables/1230204979835502592/values

```
{
  "to_add": [ {
    "id": "178101",
    "code_table_id": "1230204979835502592",
    "ordinal": 1,
    "name_en": "code",
    "name_ch": "编码",
    "description": "",
    "data_type": "STRING",
    "domain_type": null,
    "data_type_extend": null,
    "is_unique_key": false,
    "code_table_field_values": [ {
      "fd_id": "178101",
      "fd_value": "4",
      "ordinal": 4
    } ],
    "count_field_values": 3
  }, {
    "id": "178102",
    "code_table_id": "1230204979835502592",
    "ordinal": 2,
    "name_en": "province",
    "name_ch": "省",
    "description": "",
    "data_type": "STRING",
    "domain_type": null,
    "data_type_extend": null,
    "is_unique_key": false,
    "code_table_field_values": [ {
      "fd_id": "178102",
      "fd_value": "重庆市",
      "ordinal": 4
    } ],
    "count_field_values": 3
  }, {
    "id": "178103",
    "code_table_id": "1230204979835502592",
    "ordinal": 3,
    "name_en": "city",
    "name_ch": "市",
    "description": "",
    "data_type": "STRING",
    "domain_type": null,
    "data_type_extend": null,
    "is_unique_key": false,
    "code_table_field_values": [ {
      "fd_id": "178103",
      "fd_value": "重庆市",

```

```
    "ordinal" : 4
  }],
  "count_field_values" : 3
}, {
  "id" : "178104",
  "code_table_id" : "1230204979835502592",
  "ordinal" : 4,
  "name_en" : "county",
  "name_ch" : "县",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ {
    "fd_id" : "178104",
    "fd_value" : "渝东区",
    "ordinal" : 4
  } ],
  "count_field_values" : 3
} ] ]
}
```

## 响应示例

**状态码: 200**

Success, 返回数据是CodeTableFieldVO数组。

```
{
  "data" : {
    "value" : [ {
      "id" : "178101",
      "code_table_id" : "1230204979835502592",
      "ordinal" : 1,
      "name_en" : "code",
      "name_ch" : "编码",
      "description" : "",
      "data_type" : "STRING",
      "domain_type" : null,
      "data_type_extend" : null,
      "is_unique_key" : false,
      "code_table_field_values" : [ {
        "id" : "19983446",
        "fd_id" : "178101",
        "fd_value" : "1",
        "ordinal" : 1,
        "description" : null
      } ], {
        "id" : "19983450",
        "fd_id" : "178101",
        "fd_value" : "2",
        "ordinal" : 2,
        "description" : null
      } ], {
        "id" : "19983454",
        "fd_id" : "178101",
        "fd_value" : "3",
        "ordinal" : 3,
        "description" : null
      } ], {
        "id" : "19983554",
        "fd_id" : "178101",
        "fd_value" : "4",
        "ordinal" : 4,
        "description" : null
      } ],
      "count_field_values" : 4
    } ], {
  }
```

```

"id": "178102",
"code_table_id": "1230204979835502592",
"ordinal": 2,
"name_en": "province",
"name_ch": "省",
"description": "",
"data_type": "STRING",
"domain_type": null,
"data_type_extend": null,
"is_unique_key": false,
"code_table_field_values": [ {
  "id": "19983447",
  "fd_id": "178102",
  "fd_value": "安徽省",
  "ordinal": 1,
  "description": null
}, {
  "id": "19983451",
  "fd_id": "178102",
  "fd_value": "江苏省",
  "ordinal": 2,
  "description": null
}, {
  "id": "19983455",
  "fd_id": "178102",
  "fd_value": "上海市",
  "ordinal": 3,
  "description": null
}, {
  "id": "19983555",
  "fd_id": "178102",
  "fd_value": "重庆市",
  "ordinal": 4,
  "description": null
} ],
"count_field_values": 4
}, {
  "id": "178103",
  "code_table_id": "1230204979835502592",
  "ordinal": 3,
  "name_en": "city",
  "name_ch": "市",
  "description": "",
  "data_type": "STRING",
  "domain_type": null,
  "data_type_extend": null,
  "is_unique_key": false,
  "code_table_field_values": [ {
    "id": "19983448",
    "fd_id": "178103",
    "fd_value": "合肥市",
    "ordinal": 1,
    "description": null
  }, {
    "id": "19983452",
    "fd_id": "178103",
    "fd_value": "南京市",
    "ordinal": 2,
    "description": null
  }, {
    "id": "19983456",
    "fd_id": "178103",
    "fd_value": "上海市",
    "ordinal": 3,
    "description": null
  }, {
    "id": "19983556",
    "fd_id": "178103",
    "fd_value": "重庆市",

```

```
"ordinal" : 4,
  "description" : null
}],
"count_field_values" : 4
}, {
  "id" : "178104",
  "code_table_id" : "1230204979835502592",
  "ordinal" : 4,
  "name_en" : "county",
  "name_ch" : "县",
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : null,
  "data_type_extend" : null,
  "is_unique_key" : false,
  "code_table_field_values" : [ {
    "id" : "19983449",
    "fd_id" : "178104",
    "fd_value" : "瑶海区",
    "ordinal" : 1,
    "description" : null
  }, {
    "id" : "19983453",
    "fd_id" : "178104",
    "fd_value" : "雨花台区",
    "ordinal" : 2,
    "description" : null
  }, {
    "id" : "19983457",
    "fd_id" : "178104",
    "fd_value" : "浦东新区",
    "ordinal" : 3,
    "description" : null
  }, {
    "id" : "19983557",
    "fd_id" : "178104",
    "fd_value" : "渝东区",
    "ordinal" : 4,
    "description" : null
  }
  ],
  "count_field_values" : 4
}
}]
}
```

#### 状态码：400

##### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

##### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

##### Forbidden

```
{
  "error_code" : "DS.60xx",
```

```
"error_msg" : "The user does not have permission to call this API."  
}
```

**状态码：404**

Not Found

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，编辑码表的字段值，包括码表字段的新增。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class UpdateCodeTableValuesSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateCodeTableValuesRequest request = new UpdateCodeTableValuesRequest();  
        request.withId("{id}");  
        CodeTableFieldValueUpdateVO body = new CodeTableFieldValueUpdateVO();  
        List<CodeTableFieldValueVO> listToAddCodeTableFieldValues = new ArrayList<>();  
        listToAddCodeTableFieldValues.add(  
            new CodeTableFieldValueVO()  
                .withFdId("178104")  
                .withFdValue("渝东区")  
                .withOrdinal(4)  
        );  
        List<CodeTableFieldValueVO> listToAddCodeTableFieldValues1 = new ArrayList<>();  
        listToAddCodeTableFieldValues1.add(  

```

```
new CodeTableFieldValueVO()
    .withFdId("178103")
    .withFdValue("重庆市")
    .withOrdinal(4)
);
List<CodeTableFieldValueVO> listToAddCodeTableFieldValues2 = new ArrayList<>();
listToAddCodeTableFieldValues2.add(
    new CodeTableFieldValueVO()
        .withFdId("178102")
        .withFdValue("重庆市")
        .withOrdinal(4)
);
List<CodeTableFieldValueVO> listToAddCodeTableFieldValues3 = new ArrayList<>();
listToAddCodeTableFieldValues3.add(
    new CodeTableFieldValueVO()
        .withFdId("178101")
        .withFdValue("4")
        .withOrdinal(4)
);
List<CodeTableFieldVO> listbodyToAdd = new ArrayList<>();
listbodyToAdd.add(
    new CodeTableFieldVO()
        .withId("178101")
        .withCodeTableId("1230204979835502592")
        .withOrdinal(1)
        .withNameEn("code")
        .withNameCh("编码")
        .withDescription("")
        .withDataType("STRING")
        .withIsUniqueKey(false)
        .withCodeTableFieldValues(listToAddCodeTableFieldValues3)
        .withCountFieldValues(3)
);
listbodyToAdd.add(
    new CodeTableFieldVO()
        .withId("178102")
        .withCodeTableId("1230204979835502592")
        .withOrdinal(2)
        .withNameEn("province")
        .withNameCh("省")
        .withDescription("")
        .withDataType("STRING")
        .withIsUniqueKey(false)
        .withCodeTableFieldValues(listToAddCodeTableFieldValues2)
        .withCountFieldValues(3)
);
listbodyToAdd.add(
    new CodeTableFieldVO()
        .withId("178103")
        .withCodeTableId("1230204979835502592")
        .withOrdinal(3)
        .withNameEn("city")
        .withNameCh("市")
        .withDescription("")
        .withDataType("STRING")
        .withIsUniqueKey(false)
        .withCodeTableFieldValues(listToAddCodeTableFieldValues1)
        .withCountFieldValues(3)
);
listbodyToAdd.add(
    new CodeTableFieldVO()
        .withId("178104")
        .withCodeTableId("1230204979835502592")
        .withOrdinal(4)
        .withNameEn("county")
        .withNameCh("县")
        .withDescription("")
        .withDataType("STRING")
        .withIsUniqueKey(false)
);
```



```
        .withCodeTableFieldValues(listToAddCodeTableFieldValues)
        .withCountFieldValues(3)
    );
    body.withToAdd(listbodyToAdd);
    request.withBody(body);
    try {
        UpdateCodeTableValuesResponse response = client.updateCodeTableValues(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

根据请求参数，编辑码表的字段值，包括码表字段的新增。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateCodeTableValuesRequest()
        request.id = "{id}"
        listCodeTableFieldValuesToAdd = [
            CodeTableFieldValueVO(
                fd_id="178104",
                fd_value="渝东区",
                ordinal=4
            )
        ]
        listCodeTableFieldValuesToAdd1 = [
            CodeTableFieldValueVO(
                fd_id="178103",
                fd_value="重庆市",
                ordinal=4
            )
        ]
        listCodeTableFieldValuesToAdd2 = [
```

```
CodeTableFieldValueVO(
    fd_id="178102",
    fd_value="重庆市",
    ordinal=4
)
]
listCodeTableFieldValuesToAdd3 = [
    CodeTableFieldValueVO(
        fd_id="178101",
        fd_value="4",
        ordinal=4
    )
]
listToAddbody = [
    CodeTableFieldVO(
        id="178101",
        code_table_id="1230204979835502592",
        ordinal=1,
        name_en="code",
        name_ch="编码",
        description="",
        data_type="STRING",
        is_unique_key=False,
        code_table_field_values=listCodeTableFieldValuesToAdd3,
        count_field_values=3
    ),
    CodeTableFieldVO(
        id="178102",
        code_table_id="1230204979835502592",
        ordinal=2,
        name_en="province",
        name_ch="省",
        description="",
        data_type="STRING",
        is_unique_key=False,
        code_table_field_values=listCodeTableFieldValuesToAdd2,
        count_field_values=3
    ),
    CodeTableFieldVO(
        id="178103",
        code_table_id="1230204979835502592",
        ordinal=3,
        name_en="city",
        name_ch="市",
        description="",
        data_type="STRING",
        is_unique_key=False,
        code_table_field_values=listCodeTableFieldValuesToAdd1,
        count_field_values=3
    ),
    CodeTableFieldVO(
        id="178104",
        code_table_id="1230204979835502592",
        ordinal=4,
        name_en="county",
        name_ch="县",
        description="",
        data_type="STRING",
        is_unique_key=False,
        code_table_field_values=listCodeTableFieldValuesToAdd,
        count_field_values=3
    )
]
request.body = CodeTableFieldValueUpdateVO(
    to_add=listToAddbody
)
response = client.update_code_table_values(request)
print(response)
except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

根据请求参数，编辑码表的字段值，包括码表字段的新增。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateCodeTableValuesRequest{}
    request.Id = "{id}"
    fdIdCodeTableFieldValues:= "178104"
    fdValueCodeTableFieldValues:= "渝东区"
    ordinalCodeTableFieldValues:= int32(4)
    var listCodeTableFieldValuesToAdd = []model.CodeTableFieldValueVo{
        {
            FdId: &fdIdCodeTableFieldValues,
            FdValue: &fdValueCodeTableFieldValues,
            Ordinal: &ordinalCodeTableFieldValues,
        },
    }
    fdIdCodeTableFieldValues1:= "178103"
    fdValueCodeTableFieldValues1:= "重庆市"
    ordinalCodeTableFieldValues1:= int32(4)
    var listCodeTableFieldValuesToAdd1 = []model.CodeTableFieldValueVo{
        {
            FdId: &fdIdCodeTableFieldValues1,
            FdValue: &fdValueCodeTableFieldValues1,
            Ordinal: &ordinalCodeTableFieldValues1,
        },
    }
    fdIdCodeTableFieldValues2:= "178102"
    fdValueCodeTableFieldValues2:= "重庆市"
    ordinalCodeTableFieldValues2:= int32(4)
    var listCodeTableFieldValuesToAdd2 = []model.CodeTableFieldValueVo{
        {
            FdId: &fdIdCodeTableFieldValues2,
```

```

        FdValue: &fdValueCodeTableFieldValues2,
        Ordinal: &ordinalCodeTableFieldValues2,
    },
}
fdIdCodeTableFieldValues3:= "178101"
fdValueCodeTableFieldValues3:= "4"
ordinalCodeTableFieldValues3:= int32(4)
var listCodeTableFieldValuesToAdd3 = []model.CodeTableFieldValueVo{
    {
        FdId: &fdIdCodeTableFieldValues3,
        FdValue: &fdValueCodeTableFieldValues3,
        Ordinal: &ordinalCodeTableFieldValues3,
    },
}
idToAdd:= "178101"
codeTableIdToAdd:= "1230204979835502592"
descriptionToAdd:= ""
isUniqueKeyToAdd:= false
countFieldValuesToAdd:= int32(3)
idToAdd1:= "178102"
codeTableIdToAdd1:= "1230204979835502592"
descriptionToAdd1:= ""
isUniqueKeyToAdd1:= false
countFieldValuesToAdd1:= int32(3)
idToAdd2:= "178103"
codeTableIdToAdd2:= "1230204979835502592"
descriptionToAdd2:= ""
isUniqueKeyToAdd2:= false
countFieldValuesToAdd2:= int32(3)
idToAdd3:= "178104"
codeTableIdToAdd3:= "1230204979835502592"
descriptionToAdd3:= ""
isUniqueKeyToAdd3:= false
countFieldValuesToAdd3:= int32(3)
var listToAddbody = []model.CodeTableFieldVo{
    {
        Id: &idToAdd,
        CodeTableId: &codeTableIdToAdd,
        Ordinal: int32(1),
        NameEn: "code",
        NameCh: "编码",
        Description: &descriptionToAdd,
        DataType: "STRING",
        IsUniqueKey: &isUniqueKeyToAdd,
        CodeTableFieldValues: &listCodeTableFieldValuesToAdd3,
        CountFieldValues: &countFieldValuesToAdd,
    },
    {
        Id: &idToAdd1,
        CodeTableId: &codeTableIdToAdd1,
        Ordinal: int32(2),
        NameEn: "province",
        NameCh: "省",
        Description: &descriptionToAdd1,
        DataType: "STRING",
        IsUniqueKey: &isUniqueKeyToAdd1,
        CodeTableFieldValues: &listCodeTableFieldValuesToAdd2,
        CountFieldValues: &countFieldValuesToAdd1,
    },
    {
        Id: &idToAdd2,
        CodeTableId: &codeTableIdToAdd2,
        Ordinal: int32(3),
        NameEn: "city",
        NameCh: "市",
        Description: &descriptionToAdd2,
        DataType: "STRING",
        IsUniqueKey: &isUniqueKeyToAdd2,
        CodeTableFieldValues: &listCodeTableFieldValuesToAdd1,
    },
}

```

```

        CountFieldValues: &countFieldValuesToAdd2,
    },
    {
        Id: &idToAdd3,
        CodeTableId: &codeTableIdToAdd3,
        Ordinal: int32(4),
        NameEn: "county",
        NameCh: "县",
        Description: &descriptionToAdd3,
        DataType: "STRING",
        IsUniqueKey: &isUniqueKeyToAdd3,
        CodeTableFieldValues: &listCodeTableFieldValuesToAdd,
        CountFieldValues: &countFieldValuesToAdd3,
    },
}
}
request.Body = &model.CodeTableFieldValueUpdateVo{
    ToAdd: &listToAddbody,
}
response, err := client.UpdateCodeTableValues(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回数据是CodeTableFieldVO数组。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.6 流程架构接口

### 8.6.1 获取所有流程架构目录树

#### 功能介绍

获取所有目录树。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/design/biz/catalogs/tree

表 8-235 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-236 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-237 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-238 data

参数	参数类型	描述
value	Array of <b>BizCatalogVO</b> objects	value, 统一的返回结果的外层数据结构。

表 8-239 BizCatalogVO

参数	参数类型	描述
name	String	流程名称。
description	String	描述。
guid	String	对应资产中ID。
owner	String	责任人。
parent_id	String	父目录ID, 没有则为根目录。ID字符串。
prev_id	String	上个节点ID, 没有则为首节点。ID字符串。
next_id	String	下个节点ID, 没有则为尾节点。ID字符串。
id	String	创建时传空, 更新时必填。ID字符串。
qualified_id	String	认证ID, 自动生成。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
bizmetric_num	Integer	拥有业务指标数量, 前端不传。
children_num	Integer	拥有子流程的数量, 不包括子流程的子流程。
children	Array of <b>BizCatalogVO</b> objects	下层子目录, 只读。

**状态码：400**

**表 8-240 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-241 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-242 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

**表 8-243 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。



## 请求示例

获取所有目录树。

```
PUT https://{endpoint}/v2/{project_id}/design/biz/catalogs/tree
```

## 响应示例

**状态码：200**

Success, 返回data数据是带有子目录的目录。

```
{
  "data": {
    "value": [ {
      "name": "test_czh_02",
      "description": null,
      "guid": "cb46da58-2430-4148-ae3f-0c526143ab07",
      "owner": "{\"owners\": [\"test_uesr\"]}",
      "parent_id": null,
      "prev_id": "1193123592125169667",
      "next_id": "1200475693864079360",
      "id": "1193123592129363971",
      "qualified_id": "1193123592129363971",
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "create_time": "2024-01-06T09:27:12+08:00",
      "update_time": "2024-01-06T10:41:20+08:00",
      "bizmetric_num": null,
      "children_num": null,
      "children": null
    } ]
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code": "DS.60xx",
}
```

```
"error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListCatalogTreeSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListCatalogTreeRequest request = new ListCatalogTreeRequest();  
        try {  
            ListCatalogTreeResponse response = client.listCatalogTree(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCatalogTreeRequest()
        response = client.list_catalog_tree(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCatalogTreeRequest{}
    response, err := client.ListCatalogTree(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
```

```
    fmt.Println(err)
  }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是带有子目录的目录。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.6.2 查询流程架构列表

### 功能介绍

查询流程架构列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/biz/catalogs

表 8-244 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-245 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。

参数	是否必选	参数类型	描述
owner	否	String	按负责人查询。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-246 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-247 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-248 data

参数	参数类型	描述
value	<b>value</b> object	value, 统一的返回结果的外层数据结构。

表 8-249 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>BizCatalogVO</b> objects	流程架构详情。

表 8-250 BizCatalogVO

参数	参数类型	描述
name	String	流程名称。
description	String	描述。
guid	String	对应资产中ID。
owner	String	责任人。

参数	参数类型	描述
parent_id	String	父目录ID，没有则为根目录。ID字符串。
prev_id	String	上个节点ID，没有则为首节点。ID字符串。
next_id	String	下个节点ID，没有则为尾节点。ID字符串。
id	String	创建时传空，更新时必填。ID字符串。
qualified_id	String	认证ID，自动生成。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
bizmetric_num	Integer	拥有业务指标数量，前端不传。
children_num	Integer	拥有子流程的数量，不包括子流程的子流程。
children	Array of BizCatalogVO objects	下层子目录，只读。

状态码：400

表 8-251 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-252 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-253 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-254 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询流程设计中流程列表。

POST https://{endpoint}/v2/{project\_id}/design/biz/catalogs

## 响应示例

状态码：200

Success，返回data数据是BizCatalogVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "name": "test_czh_l2_01",
        "description": "aaa",

```



```
"guid" : "a43bf2af-0449-4748-bdc0-f500e1d82921",
"owner" : "{\"owners\": [\"test_uesr\"]}",
"parent_id" : "1193123592125169667",
"prev_id" : null,
"next_id" : null,
"id" : "1193132546607693824",
"qualified_id" : "1193123592125169667.1193132546607693824",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-01-06T10:02:46+08:00",
"update_time" : "2024-01-06T10:02:46+08:00",
"bizmetric_num" : null,
"children_num" : null,
"children" : null
}]
}
}
```

#### 状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

#### 状态码：404

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchCatalogsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        SearchCatalogsRequest request = new SearchCatalogsRequest();
        try {
            SearchCatalogsResponse response = client.searchCatalogs(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
```

```

        .build()

    try:
        request = SearchCatalogsRequest()
        response = client.search_catalogs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchCatalogsRequest{}
    response, err := client.SearchCatalogs(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是BizCatalogVO数组，以及总数。

状态码	描述
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.6.3 创建流程架构

### 功能介绍

创建流程架构。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/biz/catalogs

表 8-255 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-256 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-257 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	流程名称。
description	否	String	描述。
guid	否	String	对应资产中ID。
owner	是	String	责任人。
parent_id	否	String	父目录ID，没有则为根目录。ID字符串。
prev_id	否	String	上个节点ID，没有则为首节点。ID字符串。
next_id	否	String	下个节点ID，没有则为尾节点。ID字符串。
id	是	String	创建时传空，更新时必填。ID字符串。
qualified_id	否	String	认证ID，自动生成。
create_by	否	String	创建人。
update_by	否	String	更新人。
bizmetric_num	否	Integer	拥有业务指标数量，前端不传。
children_num	否	Integer	拥有子流程的数量，不包括子流程的子流程。

## 响应参数

状态码：200

表 8-258 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-259 data

参数	参数类型	描述
value	<b>BizCatalogVO</b> object	流程架构目录。

表 8-260 BizCatalogVO

参数	参数类型	描述
name	String	流程名称。
description	String	描述。
guid	String	对应资产中ID。
owner	String	责任人。
parent_id	String	父目录ID, 没有则为根目录。ID字符串。
prev_id	String	上个节点ID, 没有则为首节点。ID字符串。
next_id	String	下个节点ID, 没有则为尾节点。ID字符串。
id	String	创建时传空, 更新时必填。ID字符串。
qualified_id	String	认证ID, 自动生成。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
bizmetric_num	Integer	拥有业务指标数量, 前端不传。

参数	参数类型	描述
children_num	Integer	拥有子流程的数量，不包括子流程的子流程。
children	Array of BizCatalogVO objects	下层子目录，只读。

状态码：400

表 8-261 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-262 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-263 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-264 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入参数，创建流程架构。

POST https://{endpoint}/v2/{project\_id}/design/biz/catalogs

```
{
  "id" : "0",
  "name" : "test_create_biz_catalog",
  "parent_id" : null,
  "prev_id" : "1200475693864079360",
  "owner" : "{\"owners\":[\"test_uesr\"]}",
  "description" : "this is a demo."
}
```

## 响应示例

**状态码：200**

Success，返回数据是BizCatalogVO。

```
{
  "data" : {
    "value" : {
      "name" : "test_create_biz_catalog",
      "description" : "this is a demo.",
      "guid" : null,
      "owner" : "{\"owners\":[\"test_uesr\"]}",
      "parent_id" : null,
      "prev_id" : "1200475693864079360",
      "next_id" : null,
      "id" : "1230473571248308224",
      "qualified_id" : "1230473571248308224",
      "create_by" : "test_uesr",
      "update_by" : "test_uesr",
      "create_time" : "2024-04-18T11:02:40.577+08:00",
      "update_time" : "2024-04-18T11:02:40.577+08:00",
      "bizmetric_num" : null,
      "children_num" : null,
      "children" : null
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**



### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

### Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据传入参数，创建流程架构。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateCatalogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateCatalogRequest request = new CreateCatalogRequest();
```

```
BizCatalogVO body = new BizCatalogVO();
body.withId("0");
body.withPrevId("1200475693864079360");
body.withOwner("{\"owners\": [\"test_uesr\"]}");
body.withDescription("this is a demo.");
body.withName("test_create_biz_catalog");
request.withBody(body);
try {
    CreateCatalogResponse response = client.createCatalog(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据传入参数，创建流程架构。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateCatalogRequest()
        request.body = BizCatalogVO(
            id="0",
            prev_id="1200475693864079360",
            owner="{\"owners\": [\"test_uesr\"]}",
            description="this is a demo.",
            name="test_create_biz_catalog"
        )
        response = client.create_catalog(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据传入参数，创建流程架构。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateCatalogRequest{}
    prevIdBizCatalogVo := "1200475693864079360"
    descriptionBizCatalogVo := "this is a demo."
    request.Body = &model.BizCatalogVo{
        Id: "0",
        Previd: &prevIdBizCatalogVo,
        Owner: "{\"owners\": [\"test_uesr\"]}",
        Description: &descriptionBizCatalogVo,
        Name: "test_create_biz_catalog",
    }
    response, err := client.CreateCatalog(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回数据是BizCatalogVO。

状态码	描述
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.6.4 修改流程架构

### 功能介绍

修改流程架构。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/biz/catalogs

表 8-265 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-266 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-267 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	流程名称。
description	否	String	描述。
guid	否	String	对应资产中ID。
owner	是	String	责任人。
parent_id	否	String	父目录ID，没有则为根目录。ID字符串。
prev_id	否	String	上个节点ID，没有则为首节点。ID字符串。
next_id	否	String	下个节点ID，没有则为尾节点。ID字符串。
id	是	String	创建时传空，更新时必填。ID字符串。
qualified_id	否	String	认证ID，自动生成。
create_by	否	String	创建人。
update_by	否	String	更新人。
bizmetric_num	否	Integer	拥有业务指标数量，前端不传。
children_num	否	Integer	拥有子流程的数量，不包括子流程的子流程。

## 响应参数

状态码：200

表 8-268 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-269 data

参数	参数类型	描述
value	<b>BizCatalogVO</b> object	流程架构目录。

表 8-270 BizCatalogVO

参数	参数类型	描述
name	String	流程名称。
description	String	描述。
guid	String	对应资产中ID。
owner	String	责任人。
parent_id	String	父目录ID, 没有则为根目录。ID字符串。
prev_id	String	上个节点ID, 没有则为首节点。ID字符串。
next_id	String	下个节点ID, 没有则为尾节点。ID字符串。
id	String	创建时传空, 更新时必填。ID字符串。
qualified_id	String	认证ID, 自动生成。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
bizmetric_num	Integer	拥有业务指标数量, 前端不传。

参数	参数类型	描述
children_num	Integer	拥有子流程的数量，不包括子流程的子流程。
children	Array of BizCatalogVO objects	下层子目录，只读。

状态码：400

表 8-271 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-272 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-273 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-274 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入参数，修改流程架构。

```
PUT https://{endpoint}/v2/{project_id}/design/biz/catalogs
```

```
{
  "name": "test_czh_l2_01",
  "parent_id": "1193123592125169667",
  "prev_id": null,
  "id": "1193132546607693824",
  "owner": "{\"owners\": [\"test_uesr\"]}",
  "description": "aaa"
}
```

## 响应示例

**状态码：200**

Success，返回数据是BizCatalogVO。

```
{
  "data": {
    "value": {
      "name": "test_czh_l2_01",
      "description": "aaa",
      "guid": null,
      "owner": "{\"owners\": [\"test_uesr\"]}",
      "parent_id": "1193123592125169667",
      "prev_id": null,
      "next_id": null,
      "id": "1193132546607693824",
      "qualified_id": null,
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "create_time": "2024-01-06T10:02:46+08:00",
      "update_time": "2024-04-18T11:47:07.035+08:00",
      "bizmetric_num": null,
      "children_num": null,
      "children": null
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**



### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

### Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据传入参数，修改流程架构。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ChangeCatalogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ChangeCatalogRequest request = new ChangeCatalogRequest();
```

```
BizCatalogVO body = new BizCatalogVO();
body.withId("1193132546607693824");
body.withParentId("1193123592125169667");
body.withOwner("{\"owners\":[\"test_uesr\"]}");
body.withDescription("aaa");
body.withName("test_czh_l2_01");
request.withBody(body);
try {
    ChangeCatalogResponse response = client.changeCatalog(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据传入参数，修改流程架构。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ChangeCatalogRequest()
        request.body = BizCatalogVO(
            id="1193132546607693824",
            parent_id="1193123592125169667",
            owner="{\"owners\":[\"test_uesr\"]}",
            description="aaa",
            name="test_czh_l2_01"
        )
        response = client.change_catalog(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据传入参数，修改流程架构。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeCatalogRequest{}
    parentIdBizCatalogVo := "1193123592125169667"
    descriptionBizCatalogVo := "aaa"
    request.Body = &model.BizCatalogVo{
        Id: "1193132546607693824",
        ParentId: &parentIdBizCatalogVo,
        Owner: "{owners:[\"test_uesr\"]}",
        Description: &descriptionBizCatalogVo,
        Name: "test_czh_l2_01",
    }
    response, err := client.ChangeCatalog(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回数据是BizCatalogVO。

状态码	描述
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.6.5 删除流程架构

### 功能介绍

删除流程架构。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/biz/catalogs

表 8-275 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-276 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-277 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-278 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-279 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-280 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-281 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-282 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-283 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入参数，删除流程架构。

```
DELETE https://{endpoint}/v2/{project_id}/design/biz/catalogs
```

```
{  
  "ids" : [ "1193132546607693824" ]  
}
```

## 响应示例

### 状态码：200

Success, 返回data数据是删除个数。

```
{  
  "data" : {  
    "value" : 1  
  }  
}
```

### 状态码：400

BadRequest

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user request is illegal."  
}
```

### 状态码：401

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

### 状态码：403

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

### 状态码：404

Not Found

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据传入参数，删除流程架构。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteCatalogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteCatalogRequest request = new DeleteCatalogRequest();
        IdsParam body = new IdsParam();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("1193132546607693824");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            DeleteCatalogResponse response = client.deleteCatalog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据传入参数，删除流程架构。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```



```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteCatalogRequest()
    listIdsbody = [
        "1193132546607693824"
    ]
    request.body = IdsParam(
        ids=listIdsbody
    )
    response = client.delete_catalog(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据传入参数，删除流程架构。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteCatalogRequest{}
    var listIdsbody = []string{
        "1193132546607693824",
```

```

}
request.Body = &model.IdsParam{
    Ids: listIdsbody,
}
response, err := client.DeleteCatalog(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.6.6 查找流程架构详情

### 功能介绍

查找流程架构详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/biz/catalogs/{id}

表 8-284 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

## 请求参数

表 8-285 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-286 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-287 data

参数	参数类型	描述
value	<a href="#">BizCatalogVO</a> object	流程架构目录。

表 8-288 BizCatalogVO

参数	参数类型	描述
name	String	流程名称。
description	String	描述。
guid	String	对应资产中ID。
owner	String	责任人。
parent_id	String	父目录ID，没有则为根目录。ID字符串。
prev_id	String	上个节点ID，没有则为首节点。ID字符串。
next_id	String	下个节点ID，没有则为尾节点。ID字符串。
id	String	创建时传空，更新时必填。ID字符串。
qualified_id	String	认证ID，自动生成。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
bizmetric_num	Integer	拥有业务指标数量，前端不传。
children_num	Integer	拥有子流程的数量，不包括子流程的子流程。
children	Array of BizCatalogVO objects	下层子目录，只读。

状态码：400

表 8-289 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-290 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-291 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

**表 8-292 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据流程ID，查询流程设计中流程详情。

GET [https://{endpoint}/v2/{project\\_id}/design/biz/catalogs/904051952785162240](https://{endpoint}/v2/{project_id}/design/biz/catalogs/904051952785162240)

## 响应示例

### 状态码：200

Success, 返回data数据是BizCatalogVO详情。

```
{
  "data": {
    "value": {
      "name": "xx",
      "description": "",
      "guid": "4c9c3574-5cc7-46af-8f67-34782d5a6b32",
      "owner": "{\"owners\": [\"abc\"]}",
      "parent_id": "887366192090767360",
      "prev_id": null,
      "next_id": null,
      "id": "904051952785162240",
      "qualified_id": "887366192090767360.904051952785162240",
      "create_by": "abc",
      "update_by": "abc",
      "create_time": "2021-10-30T16:59:42+08:00",
      "update_time": "2021-10-30T16:59:42+08:00",
      "bizmetric_num": null,
      "children_num": null,
      "children": null
    }
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

### 状态码：404

Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowBizCatalogDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowBizCatalogDetailRequest request = new ShowBizCatalogDetailRequest();
        request.withId("{id}");
        try {
            ShowBizCatalogDetailResponse response = client.showBizCatalogDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowBizCatalogDetailRequest()
    request.id = "{id}"
    response = client.show_biz_catalog_detail(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBizCatalogDetailRequest{}
    request.Id = "{id}"
    response, err := client.ShowBizCatalogDetail(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```



## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是BizCatalogVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.7 数据标准模板接口

### 8.7.1 查询数据标准模板

#### 功能介绍

查询当前工作空间下的数据标准模板。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/standards/templates

表 8-293 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-294 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。

参数	是否必选	参数类型	描述
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-295 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-296 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	接口返回的数据。

表 8-297 data

参数	参数类型	描述
value	<b>value</b> object	查询数据标准模板结果

表 8-298 value

参数	参数类型	描述
allFields	Array of <a href="#">StandElementFieldVO</a> objects	数据标准全部属性，集合中是单个 StandElementFieldVO对象
optional	Array of <a href="#">StandElementFieldVO</a> objects	可选项,集合中是单个 StandElementFieldVO对象
system_default	Array of <a href="#">StandElementFieldVO</a> objects	系统默认项，集合中是单个 StandElementFieldVO对象
custom	Array of <a href="#">StandElementFieldVO</a> objects	自定义项,集合中是单个 StandElementFieldVO对象
hasTemplate	Boolean	是否使用模板

表 8-299 StandElementFieldVO

参数	参数类型	描述
fd_name	String	属性名称。
fd_name_en	String	属性英文名称。
description	String	属性描述。
descriptionEn	String	属性英文描述。
label	String	属性标签。
disabled	Boolean	是否禁用。
id	String	数据标准ID，ID字符串。
actived	Boolean	是否显示，系统默认项必然显示不允许修改。true表示使用数据标准时体现（增改查的时候可以操作该属性），false表示使用数据标准时不体现。
required	Boolean	是否必填。true：必填，false：非必填。
searchable	Boolean	是否可搜索。true表示在数据标准列表页面可搜索，false表示在数据标准列表页面不可搜索。
optional_values	String	允许值。
field_type	Integer	字段类型，0表示系统字段，1表示自定义字段。

参数	参数类型	描述
displayed_name	String	前端展示名。
displayed_name_en	String	前端展示名英文。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

状态码：400

表 8-300 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-301 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-302 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-303 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询当前工作空间下的数据标准模板。

GET https://{endpoint}/v2/{project\_id}/design/standards/templates

## 响应示例

状态码：200

Success，返回data数据是StandElementFieldVO列表。

```
{
  "data": {
    "value": {
      "allFields": [ {
        "fd_name": "nameCh",
        "fd_name_en": null,
        "description": "标准名称",
        "id": "1020622096960831488",
        "actived": true,
        "required": true,
        "searchable": true,
        "optional_values": null,
        "field_type": null,
        "displayed_name": null,
        "displayed_name_en": null,
        "create_time": "2022-09-17T09:07:50+08:00",
        "update_time": "2024-03-13T16:48:56+08:00",
        "create_by": "test_uesr",
        "update_by": "test_uesr"
      }, {
        "fd_name": "nameEn",
        "fd_name_en": null,
        "description": "标准编码",
        "id": "1020622096985997312",
```

```

"activated" : true,
"required" : true,
"searchable" : true,
"optional_values" : null,
"field_type" : null,
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
"fd_name" : "dataType",
"fd_name_en" : null,
"description" : "数据类型",
"id" : "1020622097006968832",
"activated" : true,
"required" : true,
"searchable" : true,
"optional_values" : null,
"field_type" : null,
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
"fd_name" : "dataLength",
"fd_name_en" : null,
"description" : "数据长度",
"id" : "1020622097032134656",
"activated" : true,
"required" : false,
"searchable" : false,
"optional_values" : null,
"field_type" : null,
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
"fd_name" : "hasAllowValueList",
"fd_name_en" : null,
"description" : "是否有允许值",
"id" : "1020622097048911872",
"activated" : true,
"required" : false,
"searchable" : false,
"optional_values" : null,
"field_type" : null,
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
"fd_name" : "allowList",
"fd_name_en" : null,
"description" : "允许值",
"id" : "1020622097065689088",
"activated" : true,
"required" : false,
"searchable" : false,
"optional_values" : null,
"field_type" : null,

```

```
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
  "fd_name" : "referCodeTable",
  "fd_name_en" : null,
  "description" : "引用码表",
  "id" : "1020622097086660608",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "codeStandColumn",
  "fd_name_en" : null,
  "description" : "码表字段",
  "id" : "1020622097103437824",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "dqcRule",
  "fd_name_en" : null,
  "description" : "质量规则",
  "id" : "1020622097124409344",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "ruleOwner",
  "fd_name_en" : null,
  "description" : "业务规则责任人",
  "id" : "1020622097141186560",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
}
```

```
"update_by" : "test_uesr"
}, {
  "fd_name" : "dataMonitorOwner",
  "fd_name_en" : null,
  "description" : "数据监控责任人",
  "id" : "1020622097162158080",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "standardLevel",
  "fd_name_en" : null,
  "description" : "标准层级",
  "id" : "1020622097178935296",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "description",
  "fd_name_en" : null,
  "description" : "描述",
  "id" : "1020622097195712512",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "englishName",
  "fd_name_en" : null,
  "description" : "英文名称",
  "id" : "1185628711836360704",
  "activated" : true,
  "required" : false,
  "searchable" : true,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2023-12-16T17:05:13+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}],
"custom" : [ ],
"optional" : [ {
  "fd_name" : "dataLength",
```



```

"fd_name_en" : null,
"description" : "数据长度",
"id" : "1020622097032134656",
"actived" : true,
"required" : false,
"searchable" : false,
"optional_values" : null,
"field_type" : null,
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
"fd_name" : "hasAllowValueList",
"fd_name_en" : null,
"description" : "是否有允许值",
"id" : "1020622097048911872",
"actived" : true,
"required" : false,
"searchable" : false,
"optional_values" : null,
"field_type" : null,
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
"fd_name" : "allowList",
"fd_name_en" : null,
"description" : "允许值",
"id" : "1020622097065689088",
"actived" : true,
"required" : false,
"searchable" : false,
"optional_values" : null,
"field_type" : null,
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
"fd_name" : "referCodeTable",
"fd_name_en" : null,
"description" : "引用码表",
"id" : "1020622097086660608",
"actived" : true,
"required" : false,
"searchable" : false,
"optional_values" : null,
"field_type" : null,
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
"fd_name" : "codeStandColumn",
"fd_name_en" : null,
"description" : "码表字段",
"id" : "1020622097103437824",
"actived" : true,
"required" : false,

```

```
"searchable": false,
"optional_values": null,
"field_type": null,
"displayed_name": null,
"displayed_name_en": null,
"create_time": "2022-09-17T09:07:50+08:00",
"update_time": "2024-03-13T16:48:56+08:00",
"create_by": "test_uesr",
"update_by": "test_uesr"
}, {
  "fd_name": "dqcRule",
  "fd_name_en": null,
  "description": "质量规则",
  "id": "1020622097124409344",
  "activated": true,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2022-09-17T09:07:50+08:00",
  "update_time": "2024-03-13T16:48:56+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}, {
  "fd_name": "ruleOwner",
  "fd_name_en": null,
  "description": "业务规则责任人",
  "id": "1020622097141186560",
  "activated": true,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2022-09-17T09:07:50+08:00",
  "update_time": "2024-03-13T16:48:56+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}, {
  "fd_name": "dataMonitorOwner",
  "fd_name_en": null,
  "description": "数据监控责任人",
  "id": "1020622097162158080",
  "activated": true,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2022-09-17T09:07:50+08:00",
  "update_time": "2024-03-13T16:48:56+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}, {
  "fd_name": "standardLevel",
  "fd_name_en": null,
  "description": "标准层级",
  "id": "1020622097178935296",
  "activated": true,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
```

```
"create_time" : "2022-09-17T09:07:50+08:00",
"update_time" : "2024-03-13T16:48:56+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
  "fd_name" : "description",
  "fd_name_en" : null,
  "description" : "描述",
  "id" : "1020622097195712512",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "englishName",
  "fd_name_en" : null,
  "description" : "英文名称",
  "id" : "1185628711836360704",
  "activated" : true,
  "required" : false,
  "searchable" : true,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2023-12-16T17:05:13+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}],
"hasTemplate" : true,
"system_default" : [ {
  "fd_name" : "nameCh",
  "fd_name_en" : null,
  "description" : "标准名称",
  "id" : "1020622096960831488",
  "activated" : true,
  "required" : true,
  "searchable" : true,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "nameEn",
  "fd_name_en" : null,
  "description" : "标准编码",
  "id" : "1020622096985997312",
  "activated" : true,
  "required" : true,
  "searchable" : true,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
```

```
"update_by" : "test_uesr"
}, {
  "fd_name" : "dataType",
  "fd_name_en" : null,
  "description" : "数据类型",
  "id" : "1020622097006968832",
  "activated" : true,
  "required" : true,
  "searchable" : true,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2022-09-17T09:07:50+08:00",
  "update_time" : "2024-03-13T16:48:56+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}]
}
}
```

### 状态码：400

#### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

#### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

#### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

### 状态码：404

#### Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowStandardTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))
            .build();
        ShowStandardTemplateRequest request = new ShowStandardTemplateRequest();
        try {
            ShowStandardTemplateResponse response = client.showStandardTemplate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowStandardTemplateRequest()
    response = client.show_standard_template(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowStandardTemplateRequest{}
    response, err := client.ShowStandardTemplate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是StandElementFieldVO列表。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.7.2 创建数据标准模板

### 功能介绍

创建当前工作空间下的数据标准模板自定义项。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/standards/templates

表 8-304 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-305 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 ( 响应消息头中X-Subject-Token的值) 使用Token认证时必选。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-306 请求 Body 参数

参数	是否必选	参数类型	描述
fd_name	是	String	属性名称。
fd_name_en	否	String	属性英文名称。
description	否	String	属性描述。
descriptionEn	否	String	属性英文描述。
label	否	String	属性标签。
disabled	否	Boolean	是否禁用。
id	否	String	数据标准ID，ID字符串。
actived	是	Boolean	是否显示，系统默认项必然显示不允许修改。true表示使用数据标准时体现（增改查的时候可以操作该属性），false表示使用数据标准时不体现。
required	否	Boolean	是否必填。true：必填，false：非必填。
searchable	否	Boolean	是否可搜索。true表示在数据标准列表页面可搜索，false表示在数据标准列表页面不可搜索。
optional_values	否	String	允许值。
field_type	否	Integer	字段类型，0表示系统字段，1表示自定义字段。
displayed_name	否	String	前端展示名。



参数	是否必选	参数类型	描述
displayed_name_en	否	String	前端展示名英文。
create_by	否	String	创建人。
update_by	否	String	更新人。

## 响应参数

状态码：200

表 8-307 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-308 data

参数	参数类型	描述
value	<b>StandElementFieldVO</b> object	value，统一的返回结果的外层数据结构。

表 8-309 StandElementFieldVO

参数	参数类型	描述
fd_name	String	属性名称。
fd_name_en	String	属性英文名称。
description	String	属性描述。
descriptionEn	String	属性英文描述。
label	String	属性标签。
disabled	Boolean	是否禁用。
id	String	数据标准ID，ID字符串。
actived	Boolean	是否显示，系统默认项必然显示不允许修改。true表示使用数据标准时体现（增改查的时候可以操作该属性），false表示使用数据标准时不体现。

参数	参数类型	描述
required	Boolean	是否必填。true: 必填, false: 非必填。
searchable	Boolean	是否可搜索。true表示在数据标准列表页面可搜索, false表示在数据标准列表页面不可搜索。
optional_values	String	允许值。
field_type	Integer	字段类型, 0表示系统字段, 1表示自定义字段。
displayed_name	String	前端展示名。
displayed_name_en	String	前端展示名英文。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

#### 状态码: 400

表 8-310 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

#### 状态码: 401

表 8-311 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-312 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-313 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

创建当前工作空间下的数据标准模板自定义项。

POST https://{endpoint}/v2/{project\_id}/design/standards/templates

```
{
  "fd_name": "自定义项三",
  "fd_name_en": "CustomField3",
  "description": "自定义项三",
  "id": "",
  "actived": true,
  "required": false,
  "searchable": false,
  "optional_values": "",
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2024-04-18T16:37:50+08:00",
  "update_time": "2024-04-26T14:21:18+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}
```

## 响应示例

### 状态码：200

Success

```
{
  "data": {
    "value": {
      "fd_name": "自定义项三",
      "fd_name_en": "CustomField3",
      "description": "自定义项三",
      "id": "1233446517415743488",
      "activated": true,
      "required": false,
      "searchable": false,
      "optional_values": "",
      "field_type": null,
      "displayed_name": null,
      "displayed_name_en": null,
      "create_time": "2024-04-26T15:56:06.08+08:00",
      "update_time": "2024-04-26T15:56:06.08+08:00",
      "create_by": "test_uesr",
      "update_by": "test_uesr"
    }
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

### 状态码：404

Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

创建当前工作空间下的数据标准模板自定义项。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.time.OffsetDateTime;
import java.time.format.DateTimeFormatter;

public class CreateStandardTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateStandardTemplateRequest request = new CreateStandardTemplateRequest();
        StandElementFieldVO body = new StandElementFieldVO();
        body.withUpdateBy("test_uesr");
        body.withCreateBy("test_uesr");
        body.withUpdateTime(OffsetDateTime.parse("2024-04-26T14:21:18+08:00",
            DateTimeFormatter.ISO_OFFSET_DATE_TIME));
        body.withCreateTime(OffsetDateTime.parse("2024-04-18T16:37:50+08:00",
            DateTimeFormatter.ISO_OFFSET_DATE_TIME));
        body.withOptionalValues("");
        body.withSearchable(false);
        body.withRequired(false);
        body.withActivated(true);
        body.withId("");
        body.withDescription("自定义项三");
        body.withFdNameEn("CustomField3");
        body.withFdName("自定义项三");
        request.withBody(body);
        try {
            CreateStandardTemplateResponse response = client.createStandardTemplate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

## Python

创建当前工作空间下的数据标准模板自定义项。

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateStandardTemplateRequest()  
        request.body = StandElementFieldVO(  
            update_by="test_uesr",  
            create_by="test_uesr",  
            update_time="2024-04-26T14:21:18+08:00",  
            create_time="2024-04-18T16:37:50+08:00",  
            optional_values="",  
            searchable=False,  
            required=False,  
            actived=True,  
            id="",  
            description="自定义项三",  
            fd_name_en="CustomField3",  
            fd_name="自定义项三"  
        )  
        response = client.create_standard_template(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

创建当前工作空间下的数据标准模板自定义项。

```
package main  
  
import (  
    "fmt"  
    "time"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/sdktime"
```

```
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateStandardTemplateRequest{}
    updateByStandElementFieldVo := "test_uesr"
    createByStandElementFieldVo := "test_uesr"
    dateTimeUpdateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "2024-04-26T14:21:18+08:00",
time.UTC)
    updateTimeStandElementFieldVo := sdktime.SdkTime(dateTimeUpdateTime)
    dateTimeCreateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "2024-04-18T16:37:50+08:00",
time.UTC)
    createTimeStandElementFieldVo := sdktime.SdkTime(dateTimeCreateTime)
    optionalValuesStandElementFieldVo := ""
    searchableStandElementFieldVo := false
    requiredStandElementFieldVo := false
    idStandElementFieldVo := ""
    descriptionStandElementFieldVo := "自定义项三"
    fdNameEnStandElementFieldVo := "CustomField3"
    request.Body = &model.StandElementFieldVo{
        UpdateBy: &updateByStandElementFieldVo,
        CreateBy: &createByStandElementFieldVo,
        UpdateTime: &updateTimeStandElementFieldVo,
        CreateTime: &createTimeStandElementFieldVo,
        OptionalValues: &optionalValuesStandElementFieldVo,
        Searchable: &searchableStandElementFieldVo,
        Required: &requiredStandElementFieldVo,
        Activated: true,
        Id: &idStandElementFieldVo,
        Description: &descriptionStandElementFieldVo,
        FdNameEn: &fdNameEnStandElementFieldVo,
        FdName: "自定义项三",
    }
    response, err := client.CreateStandardTemplate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

### 8.7.3 修改数据标准模板

#### 功能介绍

修改数据标准模板。

#### 调用方法

请参见[如何调用API](#)。

#### URI

PUT /v2/{project\_id}/design/standards/templates

表 8-314 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 8-315 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。



参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-316 请求 Body 参数

参数	是否必选	参数类型	描述
fd_name	是	String	属性名称。
fd_name_en	否	String	属性英文名称。
description	否	String	属性描述。
descriptionEn	否	String	属性英文描述。
label	否	String	属性标签。
disabled	否	Boolean	是否禁用。
id	否	String	数据标准ID，ID字符串。
actived	是	Boolean	是否显示，系统默认项必然显示不允许修改。true表示使用数据标准时体现（增改查的时候可以操作该属性），false表示使用数据标准时不体现。
required	否	Boolean	是否必填。true：必填，false：非必填。
searchable	否	Boolean	是否可搜索。true表示在数据标准列表页面可搜索，false表示在数据标准列表页面不可搜索。
optional_values	否	String	允许值。
field_type	否	Integer	字段类型，0表示系统字段，1表示自定义字段。
displayed_name	否	String	前端展示名。

参数	是否必选	参数类型	描述
displayed_name_en	否	String	前端展示名英文。
create_by	否	String	创建人。
update_by	否	String	更新人。

## 响应参数

状态码：200

表 8-317 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-318 data

参数	参数类型	描述
value	<b>StandElementFieldVO</b> object	value，统一的返回结果的外层数据结构。

表 8-319 StandElementFieldVO

参数	参数类型	描述
fd_name	String	属性名称。
fd_name_en	String	属性英文名称。
description	String	属性描述。
descriptionEn	String	属性英文描述。
label	String	属性标签。
disabled	Boolean	是否禁用。
id	String	数据标准ID，ID字符串。
actived	Boolean	是否显示，系统默认项必然显示不允许修改。true表示使用数据标准时体现（增改查的时候可以操作该属性），false表示使用数据标准时不体现。

参数	参数类型	描述
required	Boolean	是否必填。true: 必填, false: 非必填。
searchable	Boolean	是否可搜索。true表示在数据标准列表页面可搜索, false表示在数据标准列表页面不可搜索。
optional_values	String	允许值。
field_type	Integer	字段类型, 0表示系统字段, 1表示自定义字段。
displayed_name	String	前端展示名。
displayed_name_en	String	前端展示名英文。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

**状态码: 400**

表 8-320 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码: 401**

表 8-321 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-322 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-323 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

修改当前工作空间下的数据标准模板。

PUT https://{endpoint}/v2/{project\_id}/design/standards/templates

```
{
  "fd_name": "自定义项三",
  "fd_name_en": "CustomField33",
  "description": "自定义项三",
  "id": "1233446517415743488",
  "actived": true,
  "required": false,
  "searchable": false,
  "optional_values": "",
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2024-04-26T15:56:06.08+08:00",
  "update_time": "2024-04-26T15:56:06.08+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}
```

## 响应示例

### 状态码：200

Success

```
{
  "data": {
    "value": {
      "fd_name": "自定义项三",
      "fd_name_en": "CustomField33",
      "description": "自定义项三",
      "id": "1233446517415743488",
      "activated": true,
      "required": false,
      "searchable": false,
      "optional_values": "",
      "field_type": null,
      "displayed_name": null,
      "displayed_name_en": null,
      "create_time": "2024-04-26T07:56:06.08Z",
      "update_time": "2024-04-26T07:56:06.08Z",
      "create_by": "test_uesr",
      "update_by": "0833a573fe80d5401f6dc010a775569a"
    }
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

### 状态码：404

Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

修改当前工作空间下的数据标准模板。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.time.OffsetDateTime;
import java.time.format.DateTimeFormatter;

public class UpdateStandardTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateStandardTemplateRequest request = new UpdateStandardTemplateRequest();
        StandElementFieldVO body = new StandElementFieldVO();
        body.withUpdateBy("test_uesr");
        body.withCreateBy("test_uesr");
        body.withUpdateTime(OffsetDateTime.parse("2024-04-26T15:56:06.08+08:00",
            DateTimeFormatter.ISO_OFFSET_DATE_TIME));
        body.withCreateTime(OffsetDateTime.parse("2024-04-26T15:56:06.08+08:00",
            DateTimeFormatter.ISO_OFFSET_DATE_TIME));
        body.withOptionalValues("");
        body.withSearchable(false);
        body.withRequired(false);
        body.withActivated(true);
        body.withId("1233446517415743488");
        body.withDescription("自定义项三");
        body.withFdNameEn("CustomField33");
        body.withFdName("自定义项三");
        request.withBody(body);
        try {
            UpdateStandardTemplateResponse response = client.updateStandardTemplate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

## Python

修改当前工作空间下的数据标准模板。

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UpdateStandardTemplateRequest()  
        request.body = StandElementFieldVO(  
            update_by="test_uesr",  
            create_by="test_uesr",  
            update_time="2024-04-26T15:56:06.08+08:00",  
            create_time="2024-04-26T15:56:06.08+08:00",  
            optional_values="",  
            searchable=False,  
            required=False,  
            actived=True,  
            id="1233446517415743488",  
            description="自定义项三",  
            fd_name_en="CustomField33",  
            fd_name="自定义项三"  
        )  
        response = client.update_standard_template(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

修改当前工作空间下的数据标准模板。

```
package main  
  
import (  
    "fmt"  
    "time"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/sdktime"
```

```
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateStandardTemplateRequest{
        updateByStandElementFieldVo:= "test_uesr"
        createByStandElementFieldVo:= "test_uesr"
        dateTimeUpdateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z",
"2024-04-26T15:56:06.08+08:00", time.UTC)
        updateTimeStandElementFieldVo:= sdktime.SdkTime(dateTimeUpdateTime)
        dateTimeCreateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z",
"2024-04-26T15:56:06.08+08:00", time.UTC)
        createTimeStandElementFieldVo:= sdktime.SdkTime(dateTimeCreateTime)
        optionalValuesStandElementFieldVo:= ""
        searchableStandElementFieldVo:= false
        requiredStandElementFieldVo:= false
        idStandElementFieldVo:= "1233446517415743488"
        descriptionStandElementFieldVo:= "自定义项三"
        fdNameEnStandElementFieldVo:= "CustomField33"
        request.Body = &model.StandElementFieldVo{
            UpdateBy: &updateByStandElementFieldVo,
            CreateBy: &createByStandElementFieldVo,
            UpdateTime: &updateTimeStandElementFieldVo,
            CreateTime: &createTimeStandElementFieldVo,
            OptionalValues: &optionalValuesStandElementFieldVo,
            Searchable: &searchableStandElementFieldVo,
            Required: &requiredStandElementFieldVo,
            Activated: true,
            Id: &idStandElementFieldVo,
            Description: &descriptionStandElementFieldVo,
            FdNameEn: &fdNameEnStandElementFieldVo,
            FdName: "自定义项三",
        }
    }
    response, err := client.UpdateStandardTemplate(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。



## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.7.4 删除数据标准模板

### 功能介绍

删除数据标准模板。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/standards/templates

表 8-324 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-325 Query 参数

参数	是否必选	参数类型	描述
ids	是	String	待删除的数据标准的ID集合。

## 请求参数

表 8-326 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-327 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-328 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-329 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-330 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-331 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-332 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

删除当前工作空间下的数据标准模板。

```
DELETE https://{endpoint}/v2/{project_id}/design/standards/templates  
  
{  
  "ids" : [ "1193132546607693824" ]  
}
```

## 响应示例

### 状态码：200

Success, 返回data数据是删除个数。

```
{  
  "data" : {  
    "value" : "1"  
  }  
}
```

### 状态码：400

BadRequest

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user request is illegal."  
}
```

### 状态码：401

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

### 状态码：403

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

### 状态码：404

Not Found

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

删除当前工作空间下的数据标准模板。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteStandardTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteStandardTemplateRequest request = new DeleteStandardTemplateRequest();
        try {
            DeleteStandardTemplateResponse response = client.deleteStandardTemplate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

删除当前工作空间下的数据标准模板。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteStandardTemplateRequest()
    response = client.delete_standard_template(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

删除当前工作空间下的数据标准模板。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteStandardTemplateRequest{}
    response, err := client.DeleteStandardTemplate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.7.5 初始化数据标准模板

### 功能介绍

初始化数据标准模板。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/standards/templates/action

表 8-333 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-334 Query 参数

参数	是否必选	参数类型	描述
action-id	是	String	action-id=init, 初始化数据标准模板时固定参数。

## 请求参数

表 8-335 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-336 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-337 data

参数	参数类型	描述
value	Array of <a href="#">StandElementFieldVO</a> objects	数据标准模板字段详情数组。



表 8-338 StandElementFieldVO

参数	参数类型	描述
fd_name	String	属性名称。
fd_name_en	String	属性英文名称。
description	String	属性描述。
descriptionEn	String	属性英文描述。
label	String	属性标签。
disabled	Boolean	是否禁用。
id	String	数据标准ID, ID字符串。
activated	Boolean	是否显示, 系统默认项必然显示不允许修改。true表示使用数据标准时体现(增改查的时候可以操作该属性), false表示使用数据标准时不体现。
required	Boolean	是否必填。true: 必填, false: 非必填。
searchable	Boolean	是否可搜索。true表示在数据标准列表页面可搜索, false表示在数据标准列表页面不可搜索。
optional_values	String	允许值。
field_type	Integer	字段类型, 0表示系统字段, 1表示自定义字段。
displayed_name	String	前端展示名。
displayed_name_en	String	前端展示名英文。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

状态码: 400

表 8-339 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-340 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-341 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-342 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

初始化数据标准模板。

POST https://{endpoint}/v2/{project\_id}/design/standards/templates/action?action-id=init

```
{
  "fields": [ {
    "descriptionEn": "Standard Name",
    "fd_name": "nameCh",
    "description": "标准名称",
    "required": true,
    "searchable": true,
    "activated": true,
    "label": "标准名称",
    "displayed_name": "标准名称",
    "displayed_name_en": "Standard name"
  }, {
    "descriptionEn": "Standard Code",
    "fd_name": "nameEn",
    "description": "标准编码",
    "required": true,
    "searchable": true,
    "activated": true,
    "label": "标准编码",
    "displayed_name": "标准编码",
    "displayed_name_en": "Standard code"
  }, {
    "descriptionEn": "Data Type",
    "fd_name": "dataType",
    "description": "数据类型",
    "required": true,
    "searchable": true,
    "activated": true,
    "label": "数据类型",
    "displayed_name": "数据类型",
    "displayed_name_en": "Data type"
  }, {
    "descriptionEn": "Standard English Name",
    "fd_name": "englishName",
    "description": "英文名称",
    "required": false,
    "searchable": false,
    "activated": false,
    "label": "英文名称",
    "disabled": false,
    "displayed_name": "英文名称",
    "displayed_name_en": "Name (EN)"
  }, {
    "descriptionEn": "Data Length",
    "fd_name": "dataLength",
    "description": "数据长度",
    "required": false,
    "searchable": false,
    "activated": true,
    "label": "数据长度",
    "disabled": false,
    "displayed_name": "数据长度",
    "displayed_name_en": "Data length"
  }, {
    "descriptionEn": "Allowed Value",
    "fd_name": "hasAllowValueList",
    "description": "是否有允许值",
    "required": false,
    "searchable": false,
    "activated": false,
    "label": "是否有允许值",
    "disabled": false,
    "displayed_name": "是否有允许值",
    "displayed_name_en": "Allowed value exist"
  }, {
    "descriptionEn": "Allowed Value List",
    "fd_name": "allowList",
```

```

"description" : "允许值",
"required" : false,
"searchable" : false,
"actived" : false,
"label" : "允许值",
"disabled" : true,
"displayed_name" : "允许值",
"displayed_name_en" : "Allowed values"
}, {
"descriptionEn" : "Referenced Lookup Table",
"fd_name" : "referCodeTable",
"description" : "引用码表",
"required" : false,
"searchable" : false,
"actived" : false,
"label" : "引用码表",
"disabled" : false,
"displayed_name" : "引用码表",
"displayed_name_en" : "Lookup table"
}, {
"descriptionEn" : "Lookup Table Field",
"fd_name" : "codeStandColumn",
"description" : "码表字段",
"required" : false,
"searchable" : false,
"actived" : false,
"label" : "码表字段",
"disabled" : true,
"displayed_name" : "码表字段",
"displayed_name_en" : "Lookup table field"
}, {
"descriptionEn" : "Quality Rule",
"fd_name" : "dqcRule",
"description" : "质量规则",
"required" : false,
"searchable" : false,
"actived" : false,
"label" : "质量规则",
"disabled" : false,
"displayed_name" : "质量规则",
"displayed_name_en" : "Quality rule"
}, {
"descriptionEn" : "Owner of Business Rules",
"fd_name" : "ruleOwner",
"description" : "业务规则责任人",
"required" : false,
"searchable" : false,
"actived" : false,
"label" : "业务规则责任人",
"disabled" : false,
"displayed_name" : "业务规则责任人",
"displayed_name_en" : "Rule designer"
}, {
"descriptionEn" : "Owner of Data Monitoring",
"fd_name" : "dataMonitorOwner",
"description" : "数据监控责任人",
"required" : false,
"searchable" : false,
"actived" : false,
"label" : "数据监控责任人",
"disabled" : false,
"displayed_name" : "数据监控责任人",
"displayed_name_en" : "Rule implementer"
}, {
"descriptionEn" : "Standard Level",
"fd_name" : "standardLevel",
"description" : "标准层级",
"required" : false,
"searchable" : false,

```

```
"activated": false,
"label": "标准层级",
"disabled": false,
"displayed_name": "标准层级",
"displayed_name_en": "Standard level"
}, {
"descriptionEn": "Description",
"fd_name": "description",
"description": "描述",
"required": false,
"searchable": false,
"activated": true,
"label": "描述",
"disabled": false,
"displayed_name": "描述",
"displayed_name_en": "Description"
}]
}
```

## 响应示例

**状态码: 200**

Success

```
{
  "data": {
    "value": [ {
      "fd_name": "nameCh",
      "fd_name_en": null,
      "description": "标准名称",
      "id": "1230921379143135232",
      "activated": true,
      "required": true,
      "searchable": true,
      "optional_values": null,
      "field_type": null,
      "displayed_name": null,
      "displayed_name_en": null,
      "create_time": "2024-04-19T16:42:06+08:00",
      "update_time": "2024-04-19T16:42:06+08:00",
      "create_by": "test_uesr",
      "update_by": "test_uesr"
    }, {
      "fd_name": "nameEn",
      "fd_name_en": null,
      "description": "标准编码",
      "id": "1230921379164106752",
      "activated": true,
      "required": true,
      "searchable": true,
      "optional_values": null,
      "field_type": null,
      "displayed_name": null,
      "displayed_name_en": null,
      "create_time": "2024-04-19T16:42:06+08:00",
      "update_time": "2024-04-19T16:42:06+08:00",
      "create_by": "test_uesr",
      "update_by": "test_uesr"
    }, {
      "fd_name": "dataType",
      "fd_name_en": null,
      "description": "数据类型",
      "id": "1230921379180883968",
      "activated": true,
      "required": true,
      "searchable": true,
      "optional_values": null,
      "field_type": null,
    }
  ]
}
```

```
"displayed_name" : null,
"displayed_name_en" : null,
"create_time" : "2024-04-19T16:42:06+08:00",
"update_time" : "2024-04-19T16:42:06+08:00",
"create_by" : "test_uesr",
"update_by" : "test_uesr"
}, {
  "fd_name" : "englishName",
  "fd_name_en" : null,
  "description" : "英文名称",
  "id" : "1230921379201855488",
  "activated" : false,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2024-04-19T16:42:06+08:00",
  "update_time" : "2024-04-19T16:42:06+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "dataLength",
  "fd_name_en" : null,
  "description" : "数据长度",
  "id" : "1230921379222827008",
  "activated" : true,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2024-04-19T16:42:06+08:00",
  "update_time" : "2024-04-19T16:42:06+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "hasAllowValueList",
  "fd_name_en" : null,
  "description" : "是否有允许值",
  "id" : "1230921379239604224",
  "activated" : false,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2024-04-19T16:42:06+08:00",
  "update_time" : "2024-04-19T16:42:06+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
}, {
  "fd_name" : "allowList",
  "fd_name_en" : null,
  "description" : "允许值",
  "id" : "1230921379260575744",
  "activated" : false,
  "required" : false,
  "searchable" : false,
  "optional_values" : null,
  "field_type" : null,
  "displayed_name" : null,
  "displayed_name_en" : null,
  "create_time" : "2024-04-19T16:42:06+08:00",
  "update_time" : "2024-04-19T16:42:06+08:00",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr"
```

```

"update_by": "test_uesr"
}, {
  "fd_name": "referCodeTable",
  "fd_name_en": null,
  "description": "引用码表",
  "id": "1230921379277352960",
  "activated": false,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2024-04-19T16:42:06+08:00",
  "update_time": "2024-04-19T16:42:06+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}, {
  "fd_name": "codeStandColumn",
  "fd_name_en": null,
  "description": "码表字段",
  "id": "1230921379298324480",
  "activated": false,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2024-04-19T16:42:06+08:00",
  "update_time": "2024-04-19T16:42:06+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}, {
  "fd_name": "dqcRule",
  "fd_name_en": null,
  "description": "质量规则",
  "id": "1230921379315101696",
  "activated": false,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2024-04-19T16:42:06+08:00",
  "update_time": "2024-04-19T16:42:06+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}, {
  "fd_name": "ruleOwner",
  "fd_name_en": null,
  "description": "业务规则责任人",
  "id": "1230921379340267520",
  "activated": false,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2024-04-19T16:42:06+08:00",
  "update_time": "2024-04-19T16:42:06+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}, {
  "fd_name": "dataMonitorOwner",
  "fd_name_en": null,
  "description": "数据监控责任人",

```

```
"id": "1230921379361239040",
"activated": false,
"required": false,
"searchable": false,
"optional_values": null,
"field_type": null,
"displayed_name": null,
"displayed_name_en": null,
"create_time": "2024-04-19T16:42:06+08:00",
"update_time": "2024-04-19T16:42:06+08:00",
"create_by": "test_uesr",
"update_by": "test_uesr"
}, {
  "fd_name": "standardLevel",
  "fd_name_en": null,
  "description": "标准层级",
  "id": "1230921379382210560",
  "activated": false,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2024-04-19T16:42:06+08:00",
  "update_time": "2024-04-19T16:42:06+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}, {
  "fd_name": "description",
  "fd_name_en": null,
  "description": "描述",
  "id": "1230921379398987776",
  "activated": true,
  "required": false,
  "searchable": false,
  "optional_values": null,
  "field_type": null,
  "displayed_name": null,
  "displayed_name_en": null,
  "create_time": "2024-04-19T16:42:06+08:00",
  "update_time": "2024-04-19T16:42:06+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr"
}]
}
```

### 状态码: 400

#### BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码: 401

#### Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码: 403

#### Forbidden



```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

初始化数据标准模板。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class InitializeStandardTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        InitializeStandardTemplateRequest request = new InitializeStandardTemplateRequest();
        StandElementFieldVOList body = new StandElementFieldVOList();
        List<StandElementFieldVO> listbodyFields = new ArrayList<>();
        listbodyFields.add(
            new StandElementFieldVO()
                .withFdName("nameCh")
                .withDescription("标准名称")
                .withDescriptionEn("Standard Name")
                .withLabel("标准名称")
                .withActivated(true)
        );
    }
}
```

```
.withRequired(true)
.withSearchable(true)
.withDisplayName("标准名称")
.withDisplayNameEn("Standard name")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("nameEn")
        .withDescription("标准编码")
        .withDescriptionEn("Standard Code")
        .withLabel("标准编码")
        .withActivated(true)
        .withRequired(true)
        .withSearchable(true)
        .withDisplayName("标准编码")
        .withDisplayNameEn("Standard code")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("dataType")
        .withDescription("数据类型")
        .withDescriptionEn("Data Type")
        .withLabel("数据类型")
        .withActivated(true)
        .withRequired(true)
        .withSearchable(true)
        .withDisplayName("数据类型")
        .withDisplayNameEn("Data type")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("englishName")
        .withDescription("英文名称")
        .withDescriptionEn("Standard English Name")
        .withLabel("英文名称")
        .withDisabled(false)
        .withActivated(false)
        .withRequired(false)
        .withSearchable(false)
        .withDisplayName("英文名称")
        .withDisplayNameEn("Name (EN)")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("dataLength")
        .withDescription("数据长度")
        .withDescriptionEn("Data Length")
        .withLabel("数据长度")
        .withDisabled(false)
        .withActivated(true)
        .withRequired(false)
        .withSearchable(false)
        .withDisplayName("数据长度")
        .withDisplayNameEn("Data length")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("hasAllowValueList")
        .withDescription("是否有允许值")
        .withDescriptionEn("Allowed Value")
        .withLabel("是否有允许值")
        .withDisabled(false)
        .withActivated(false)
        .withRequired(false)
        .withSearchable(false)
        .withDisplayName("是否有允许值")
        .withDisplayNameEn("Allowed value exist")
);
listbodyFields.add(
```

```
new StandElementFieldVO()
    .withFdName("allowList")
    .withDescription("允许值")
    .withDescriptionEn("Allowed Value List")
    .withLabel("允许值")
    .withDisabled(true)
    .withActived(false)
    .withRequired(false)
    .withSearchable(false)
    .withDisplayedName("允许值")
    .withDisplayedNameEn("Allowed values")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("referCodeTable")
        .withDescription("引用码表")
        .withDescriptionEn("Referenced Lookup Table")
        .withLabel("引用码表")
        .withDisabled(false)
        .withActived(false)
        .withRequired(false)
        .withSearchable(false)
        .withDisplayedName("引用码表")
        .withDisplayedNameEn("Lookup table")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("codeStandColumn")
        .withDescription("码表字段")
        .withDescriptionEn("Lookup Table Field")
        .withLabel("码表字段")
        .withDisabled(true)
        .withActived(false)
        .withRequired(false)
        .withSearchable(false)
        .withDisplayedName("码表字段")
        .withDisplayedNameEn("Lookup table field")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("dqcRule")
        .withDescription("质量规则")
        .withDescriptionEn("Quality Rule")
        .withLabel("质量规则")
        .withDisabled(false)
        .withActived(false)
        .withRequired(false)
        .withSearchable(false)
        .withDisplayedName("质量规则")
        .withDisplayedNameEn("Quality rule")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("ruleOwner")
        .withDescription("业务规则责任人")
        .withDescriptionEn("Owner of Business Rules")
        .withLabel("业务规则责任人")
        .withDisabled(false)
        .withActived(false)
        .withRequired(false)
        .withSearchable(false)
        .withDisplayedName("业务规则责任人")
        .withDisplayedNameEn("Rule designer")
);
listbodyFields.add(
    new StandElementFieldVO()
        .withFdName("dataMonitorOwner")
        .withDescription("数据监控责任人")
        .withDescriptionEn("Owner of Data Monitoring")
```

```
        .withLabel("数据监控责任人")
        .withDisabled(false)
        .withActivated(false)
        .withRequired(false)
        .withSearchable(false)
        .withDisplayedName("数据监控责任人")
        .withDisplayedNameEn("Rule implementer")
    );
    listbodyFields.add(
        new StandElementFieldVO()
            .withFdName("standardLevel")
            .withDescription("标准层级")
            .withDescriptionEn("Standard Level")
            .withLabel("标准层级")
            .withDisabled(false)
            .withActivated(false)
            .withRequired(false)
            .withSearchable(false)
            .withDisplayedName("标准层级")
            .withDisplayedNameEn("Standard level")
    );
    listbodyFields.add(
        new StandElementFieldVO()
            .withFdName("description")
            .withDescription("描述")
            .withDescriptionEn("Description")
            .withLabel("描述")
            .withDisabled(false)
            .withActivated(true)
            .withRequired(false)
            .withSearchable(false)
            .withDisplayedName("描述")
            .withDisplayedNameEn("Description")
    );
    body.withFields(listbodyFields);
    request.withBody(body);
    try {
        InitializeStandardTemplateResponse response = client.initializeStandardTemplate(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

初始化数据标准模板。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = InitializeStandardTemplateRequest()
    listFieldsbody = [
        StandElementFieldVO(
            fd_name="nameCh",
            description="标准名称",
            description_en="Standard Name",
            label="标准名称",
            activated=True,
            required=True,
            searchable=True,
            displayed_name="标准名称",
            displayed_name_en="Standard name"
        ),
        StandElementFieldVO(
            fd_name="nameEn",
            description="标准编码",
            description_en="Standard Code",
            label="标准编码",
            activated=True,
            required=True,
            searchable=True,
            displayed_name="标准编码",
            displayed_name_en="Standard code"
        ),
        StandElementFieldVO(
            fd_name="dataType",
            description="数据类型",
            description_en="Data Type",
            label="数据类型",
            activated=True,
            required=True,
            searchable=True,
            displayed_name="数据类型",
            displayed_name_en="Data type"
        ),
        StandElementFieldVO(
            fd_name="englishName",
            description="英文名称",
            description_en="Standard English Name",
            label="英文名称",
            disabled=False,
            activated=False,
            required=False,
            searchable=False,
            displayed_name="英文名称",
            displayed_name_en="Name (EN)"
        ),
        StandElementFieldVO(
            fd_name="dataLength",
            description="数据长度",
            description_en="Data Length",
            label="数据长度",
            disabled=False,
            activated=True,
            required=False,
```

```

searchable=False,
displayed_name="数据长度",
displayed_name_en="Data length"
),
StandElementFieldVO(
fd_name="hasAllowValueList",
description="是否有允许值",
description_en="Allowed Value",
label="是否有允许值",
disabled=False,
actived=False,
required=False,
searchable=False,
displayed_name="是否有允许值",
displayed_name_en="Allowed value exist"
),
StandElementFieldVO(
fd_name="allowList",
description="允许值",
description_en="Allowed Value List",
label="允许值",
disabled=True,
actived=False,
required=False,
searchable=False,
displayed_name="允许值",
displayed_name_en="Allowed values"
),
StandElementFieldVO(
fd_name="referCodeTable",
description="引用码表",
description_en="Referenced Lookup Table",
label="引用码表",
disabled=False,
actived=False,
required=False,
searchable=False,
displayed_name="引用码表",
displayed_name_en="Lookup table"
),
StandElementFieldVO(
fd_name="codeStandColumn",
description="码表字段",
description_en="Lookup Table Field",
label="码表字段",
disabled=True,
actived=False,
required=False,
searchable=False,
displayed_name="码表字段",
displayed_name_en="Lookup table field"
),
StandElementFieldVO(
fd_name="dqRule",
description="质量规则",
description_en="Quality Rule",
label="质量规则",
disabled=False,
actived=False,
required=False,
searchable=False,
displayed_name="质量规则",
displayed_name_en="Quality rule"
),
StandElementFieldVO(
fd_name="ruleOwner",
description="业务规则责任人",
description_en="Owner of Business Rules",
label="业务规则责任人",

```

```

        disabled=False,
        actived=False,
        required=False,
        searchable=False,
        displayed_name="业务规则责任人",
        displayed_name_en="Rule designer"
    ),
    StandElementFieldVO(
        fd_name="dataMonitorOwner",
        description="数据监控责任人",
        description_en="Owner of Data Monitoring",
        label="数据监控责任人",
        disabled=False,
        actived=False,
        required=False,
        searchable=False,
        displayed_name="数据监控责任人",
        displayed_name_en="Rule implementer"
    ),
    StandElementFieldVO(
        fd_name="standardLevel",
        description="标准层级",
        description_en="Standard Level",
        label="标准层级",
        disabled=False,
        actived=False,
        required=False,
        searchable=False,
        displayed_name="标准层级",
        displayed_name_en="Standard level"
    ),
    StandElementFieldVO(
        fd_name="description",
        description="描述",
        description_en="Description",
        label="描述",
        disabled=False,
        actived=True,
        required=False,
        searchable=False,
        displayed_name="描述",
        displayed_name_en="Description"
    )
]
request.body = StandElementFieldVOList(
    fields=listFieldsbody
)
response = client.initialize_standard_template(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

## Go

初始化数据标准模板。

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.InitializeStandardTemplateRequest{
        descriptionFields:= "标准名称"
        descriptionEnFields:= "Standard Name"
        labelFields:= "标准名称"
        requiredFields:= true
        searchableFields:= true
        displayedNameFields:= "标准名称"
        displayedNameEnFields:= "Standard name"
        descriptionFields1:= "标准编码"
        descriptionEnFields1:= "Standard Code"
        labelFields1:= "标准编码"
        requiredFields1:= true
        searchableFields1:= true
        displayedNameFields1:= "标准编码"
        displayedNameEnFields1:= "Standard code"
        descriptionFields2:= "数据类型"
        descriptionEnFields2:= "Data Type"
        labelFields2:= "数据类型"
        requiredFields2:= true
        searchableFields2:= true
        displayedNameFields2:= "数据类型"
        displayedNameEnFields2:= "Data type"
        descriptionFields3:= "英文名称"
        descriptionEnFields3:= "Standard English Name"
        labelFields3:= "英文名称"
        disabledFields:= false
        requiredFields3:= false
        searchableFields3:= false
        displayedNameFields3:= "英文名称"
        displayedNameEnFields3:= "Name (EN)"
        descriptionFields4:= "数据长度"
        descriptionEnFields4:= "Data Length"
        labelFields4:= "数据长度"
        disabledFields1:= false
        requiredFields4:= false
        searchableFields4:= false
        displayedNameFields4:= "数据长度"
        displayedNameEnFields4:= "Data length"
        descriptionFields5:= "是否有允许值"
        descriptionEnFields5:= "Allowed Value"
        labelFields5:= "是否有允许值"
        disabledFields2:= false
        requiredFields5:= false
        searchableFields5:= false
        displayedNameFields5:= "是否有允许值"
        displayedNameEnFields5:= "Allowed value exist"
        descriptionFields6:= "允许值"
```



```

descriptionEnFields6:= "Allowed Value List"
labelFields6:= "允许值"
disabledFields3:= true
requiredFields6:= false
searchableFields6:= false
displayNameFields6:= "允许值"
displayNameEnFields6:= "Allowed values"
descriptionFields7:= "引用码表"
descriptionEnFields7:= "Referenced Lookup Table"
labelFields7:= "引用码表"
disabledFields4:= false
requiredFields7:= false
searchableFields7:= false
displayNameFields7:= "引用码表"
displayNameEnFields7:= "Lookup table"
descriptionFields8:= "码表字段"
descriptionEnFields8:= "Lookup Table Field"
labelFields8:= "码表字段"
disabledFields5:= true
requiredFields8:= false
searchableFields8:= false
displayNameFields8:= "码表字段"
displayNameEnFields8:= "Lookup table field"
descriptionFields9:= "质量规则"
descriptionEnFields9:= "Quality Rule"
labelFields9:= "质量规则"
disabledFields6:= false
requiredFields9:= false
searchableFields9:= false
displayNameFields9:= "质量规则"
displayNameEnFields9:= "Quality rule"
descriptionFields10:= "业务规则责任人"
descriptionEnFields10:= "Owner of Business Rules"
labelFields10:= "业务规则责任人"
disabledFields7:= false
requiredFields10:= false
searchableFields10:= false
displayNameFields10:= "业务规则责任人"
displayNameEnFields10:= "Rule designer"
descriptionFields11:= "数据监控责任人"
descriptionEnFields11:= "Owner of Data Monitoring"
labelFields11:= "数据监控责任人"
disabledFields8:= false
requiredFields11:= false
searchableFields11:= false
displayNameFields11:= "数据监控责任人"
displayNameEnFields11:= "Rule implementer"
descriptionFields12:= "标准层级"
descriptionEnFields12:= "Standard Level"
labelFields12:= "标准层级"
disabledFields9:= false
requiredFields12:= false
searchableFields12:= false
displayNameFields12:= "标准层级"
displayNameEnFields12:= "Standard level"
descriptionFields13:= "描述"
descriptionEnFields13:= "Description"
labelFields13:= "描述"
disabledFields10:= false
requiredFields13:= false
searchableFields13:= false
displayNameFields13:= "描述"
displayNameEnFields13:= "Description"
var listFieldsbody = []model.StandElementFieldVo{
    {
        FdName: "nameCh",
        Description: &descriptionFields,
        DescriptionEn: &descriptionEnFields,
        Label: &labelFields,
    }
}

```

```

    Activated: true,
    Required: &requiredFields,
    Searchable: &searchableFields,
    DisplayName: &displayedNameFields,
    DisplayNameEn: &displayedNameEnFields,
  },
  {
    FdName: "nameEn",
    Description: &descriptionFields1,
    DescriptionEn: &descriptionEnFields1,
    Label: &labelFields1,
    Activated: true,
    Required: &requiredFields1,
    Searchable: &searchableFields1,
    DisplayName: &displayedNameFields1,
    DisplayNameEn: &displayedNameEnFields1,
  },
  {
    FdName: "dataType",
    Description: &descriptionFields2,
    DescriptionEn: &descriptionEnFields2,
    Label: &labelFields2,
    Activated: true,
    Required: &requiredFields2,
    Searchable: &searchableFields2,
    DisplayName: &displayedNameFields2,
    DisplayNameEn: &displayedNameEnFields2,
  },
  {
    FdName: "englishName",
    Description: &descriptionFields3,
    DescriptionEn: &descriptionEnFields3,
    Label: &labelFields3,
    Disabled: &disabledFields,
    Activated: false,
    Required: &requiredFields3,
    Searchable: &searchableFields3,
    DisplayName: &displayedNameFields3,
    DisplayNameEn: &displayedNameEnFields3,
  },
  {
    FdName: "dataLength",
    Description: &descriptionFields4,
    DescriptionEn: &descriptionEnFields4,
    Label: &labelFields4,
    Disabled: &disabledFields1,
    Activated: true,
    Required: &requiredFields4,
    Searchable: &searchableFields4,
    DisplayName: &displayedNameFields4,
    DisplayNameEn: &displayedNameEnFields4,
  },
  {
    FdName: "hasAllowValueList",
    Description: &descriptionFields5,
    DescriptionEn: &descriptionEnFields5,
    Label: &labelFields5,
    Disabled: &disabledFields2,
    Activated: false,
    Required: &requiredFields5,
    Searchable: &searchableFields5,
    DisplayName: &displayedNameFields5,
    DisplayNameEn: &displayedNameEnFields5,
  },
  {
    FdName: "allowList",
    Description: &descriptionFields6,
    DescriptionEn: &descriptionEnFields6,
    Label: &labelFields6,
  }

```

```

    Disabled: &disabledFields3,
    Activated: false,
    Required: &requiredFields6,
    Searchable: &searchableFields6,
    DisplayName: &displayedNameFields6,
    DisplayNameEn: &displayedNameEnFields6,
  },
  {
    FdName: "referCodeTable",
    Description: &descriptionFields7,
    DescriptionEn: &descriptionEnFields7,
    Label: &labelFields7,
    Disabled: &disabledFields4,
    Activated: false,
    Required: &requiredFields7,
    Searchable: &searchableFields7,
    DisplayName: &displayedNameFields7,
    DisplayNameEn: &displayedNameEnFields7,
  },
  {
    FdName: "codeStandColumn",
    Description: &descriptionFields8,
    DescriptionEn: &descriptionEnFields8,
    Label: &labelFields8,
    Disabled: &disabledFields5,
    Activated: false,
    Required: &requiredFields8,
    Searchable: &searchableFields8,
    DisplayName: &displayedNameFields8,
    DisplayNameEn: &displayedNameEnFields8,
  },
  {
    FdName: "dqcRule",
    Description: &descriptionFields9,
    DescriptionEn: &descriptionEnFields9,
    Label: &labelFields9,
    Disabled: &disabledFields6,
    Activated: false,
    Required: &requiredFields9,
    Searchable: &searchableFields9,
    DisplayName: &displayedNameFields9,
    DisplayNameEn: &displayedNameEnFields9,
  },
  {
    FdName: "ruleOwner",
    Description: &descriptionFields10,
    DescriptionEn: &descriptionEnFields10,
    Label: &labelFields10,
    Disabled: &disabledFields7,
    Activated: false,
    Required: &requiredFields10,
    Searchable: &searchableFields10,
    DisplayName: &displayedNameFields10,
    DisplayNameEn: &displayedNameEnFields10,
  },
  {
    FdName: "dataMonitorOwner",
    Description: &descriptionFields11,
    DescriptionEn: &descriptionEnFields11,
    Label: &labelFields11,
    Disabled: &disabledFields8,
    Activated: false,
    Required: &requiredFields11,
    Searchable: &searchableFields11,
    DisplayName: &displayedNameFields11,
    DisplayNameEn: &displayedNameEnFields11,
  },
  {
    FdName: "standardLevel",

```

```
        Description: &descriptionFields12,
        DescriptionEn: &descriptionEnFields12,
        Label: &labelFields12,
        Disabled: &disabledFields9,
        Activated: false,
        Required: &requiredFields12,
        Searchable: &searchableFields12,
        DisplayName: &displayedNameFields12,
        DisplayNameEn: &displayedNameEnFields12,
    },
    {
        FdName: "description",
        Description: &descriptionFields13,
        DescriptionEn: &descriptionEnFields13,
        Label: &labelFields13,
        Disabled: &disabledFields10,
        Activated: true,
        Required: &requiredFields13,
        Searchable: &searchableFields13,
        DisplayName: &displayedNameFields13,
        DisplayNameEn: &displayedNameEnFields13,
    },
}
request.Body = &model.StandElementFieldVoList{
    Fields: &listFieldsbody,
}
response, err := client.InitializeStandardTemplate(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8 审批管理接口

## 8.8.1 获取审批单

### 功能介绍

获取审批单。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/approvals

表 8-343 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-344 Query 参数

参数	是否必选	参数类型	描述
biz_id	否	String	业务定义ID，ID字符串。
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。
approval_status	否	String	审批单状态。 枚举值： <ul style="list-style-type: none"><li>DEVELOPING: 待审批</li><li>FINISHED: 已审批</li></ul>
approval_status_detail	否	String	审批单状态。 枚举值： <ul style="list-style-type: none"><li>DEVELOPING: 待审批</li><li>APPROVED: 审批通过</li><li>REJECT: 审批驳回</li></ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"><li>PUBLISH: 发布</li><li>OFFLINE: 下线</li></ul>

参数	是否必选	参数类型	描述
biz_type	否	String	按业务类型查询，可选业务类型有：ATOMIC_INDEX（原子指标）、DERIVATIVE_INDEX（衍生指标）、DIMENSION（维度）、TIME_CONDITION（时间限定）、DIMENSION_LOGIC_TABLE（维度表）、FACT_LOGIC_TABLE（事实表）、AGGREGATION_LOGIC_TABLE（汇总表）、TABLE_MODEL（关系建模表）、CODE_TABLE（码表）、STANDARD_ELEMENT（数据标准）、BIZ_METRIC（业务指标）、COMPOUND_METRIC（复合指标）、SUBJECT（主题）、ATOMIC_METRIC（原子指标（新））、DERIVED_METRIC（衍生指标（新））、COMPOSITE_METRIC（复合指标（新））。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-345 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-346 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-347 data

参数	参数类型	描述
value	<a href="#">value</a> object	value，统一的返回结果的外层数据结构。

表 8-348 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>ApprovalVO</b> objects	查询到的审批单对象 ( ApprovalVO ) 数组。

表 8-349 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

**状态码：400**

表 8-350 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-351 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-352 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-353 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

分页查询状态为审批中的审批单信息。

```
GET https://{endpoint}/v2/{project_id}/design/approvals?
offset=1&limit=10&approval_status=DEVELOPING&is_approval_time=false
```

## 响应示例

状态码：200

Success，返回data数据是ApprovalVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1230929439261609984",
        "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "name_ch": "test_0408",
        "name_en": "test_0408",
        "biz_id": "1226923406239399936",
        "biz_type": "TABLE_MODEL",
        "biz_info": "{}",
        "biz_info_obj": {
          "id": "1226923406239399936",
          "model_id": "1183882892003127296",
          "parent_table_id": null,
          "parent_table_name": null,
          "parent_table_code": null,
          "related_logic_table_id": null,
          "related_logic_table_name": null,
          "related_logic_table_model_id": null,
          "related_logic_table_model_name": null,
          "model": null,

```

```

"data_format": "Parquet",
"obs_bucket": null,
"obs_location": null,
"configs": "{}",
"table_type": "DWS_ROW",
"owner": "",
"tb_name": "test_0408",
"dw_id": "d2d09df46d5745ceaa0702c9245852fa",
"db_name": null,
"queue_name": null,
"schema": null,
"extend_info": "{\"compression\": \"NO\", \"dirtyOutDatabase\": \"\", \"dirtyOutPrefix\": \"\", \"dirtyOutSuffix\": \"\", \"dirtyOutSwitch\": \"false\", \"distribute\": \"HASH\"}",
"tb_guid": null,
"tb_id": null,
"logic_tb_name": "test_0408",
"logic_tb_guid": "",
"description": "无",
"status": "DRAFT",
"logic_tb_id": null,
"biz_catalog_id": "1193142949320474624",
"catalog_path": null,
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-04-08T15:55:35+08:00",
"update_time": "2024-04-17T16:20:47+08:00",
"tags": [],
"approval_info": null,
"new_biz": null,
"attributes": [ {
  "id": "1226923406327480320",
  "name_en": "aa",
  "name_ch": "aa",
  "description": "",
  "obs_location": null,
  "create_by": null,
  "update_by": null,
  "data_type": "BIGINT",
  "domain_type": "NUMBER",
  "data_type_extend": "",
  "is_primary_key": false,
  "is_partition_key": false,
  "is_foreign_key": false,
  "extend_field": false,
  "not_null": false,
  "ordinal": 1,
  "table_model_id": null,
  "create_time": "2024-04-08T15:55:35+08:00",
  "update_time": "2024-04-17T16:20:47+08:00",
  "tags": [],
  "secrecy_levels": [],
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": null,
  "alias": "",
  "self_defined_fields": [ {
    "fd_name_ch": "aa",
    "fd_name_en": "aa",
    "not_null": false,
    "fd_value": ""
  }, {
    "fd_name_ch": "bb",
    "fd_name_en": "bb",
    "not_null": false,
    "fd_value": ""
  } ],
  "code": "",
  "related_logic_attr_id": null,
  "related_logic_attr_name": null,

```

```

    "related_logic_attr_name_en" : null
  }, {
    "id" : "1226923406327480321",
    "name_en" : "bb",
    "name_ch" : "bb",
    "description" : "",
    "obs_location" : null,
    "create_by" : null,
    "update_by" : null,
    "data_type" : "BIGINT",
    "domain_type" : "NUMBER",
    "data_type_extend" : "",
    "is_primary_key" : false,
    "is_partition_key" : false,
    "is_foreign_key" : false,
    "extend_field" : false,
    "not_null" : false,
    "ordinal" : 2,
    "table_model_id" : null,
    "create_time" : "2024-04-08T15:55:35+08:00",
    "update_time" : "2024-04-17T16:20:47+08:00",
    "tags" : [ ],
    "secrecy_levels" : [ ],
    "stand_row_id" : null,
    "stand_row_name" : null,
    "quality_infos" : null,
    "alias" : "",
    "self_defined_fields" : [ {
      "fd_name_ch" : "aa",
      "fd_name_en" : "aa",
      "not_null" : false,
      "fd_value" : ""
    }, {
      "fd_name_ch" : "bb",
      "fd_name_en" : "bb",
      "not_null" : false,
      "fd_value" : ""
    }
  ],
    "code" : "",
    "related_logic_attr_id" : null,
    "related_logic_attr_name" : null,
    "related_logic_attr_name_en" : null
  } ],
  "mappings" : null,
  "relations" : [ ],
  "dw_type" : "DWS",
  "dw_name" : null,
  "l1" : null,
  "l2" : null,
  "l3" : null,
  "l1_id" : null,
  "l2_id" : null,
  "l3_id" : "1193142949320474624",
  "partition_conf" : null,
  "dlf_task_id" : null,
  "use_recently_partition" : false,
  "reversed" : null,
  "dirty_out_switch" : false,
  "dirty_out_database" : "",
  "dirty_out_prefix" : "",
  "dirty_out_suffix" : "",
  "quality_owner" : null,
  "quality_id" : null,
  "distribute" : "HASH",
  "distribute_column" : null,
  "compression" : "NO",
  "pre_combine_field" : null,
  "is_partition" : false,
  "physical_table" : "CREATE_FAILED",

```

```
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "NO_NEED",
"summary_status" : "SUMMARY_FAILED",
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE",
"alias" : null,
"self_defined_fields" : [ {
  "fd_name_ch" : "自定义项1",
  "fd_name_en" : "selfDefine1",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "自定义项2",
  "fd_name_en" : "selfDefine2",
  "not_null" : true,
  "fd_value" : "aaa"
} ],
"code" : null,
"has_related_physical_table" : false,
"has_related_logic_table" : false
},
"biz_version" : 0,
"biz_status" : "PUBLISH_DEVELOPING",
"approval_status" : "DEVELOPING",
"approval_type" : "PUBLISH",
"submit_time" : "2024-04-19T17:14:08+08:00",
"create_by" : "test_uesr",
"l1" : "test_import",
"l2" : null,
"l3" : null,
"approval_time" : "2024-04-19T17:14:08+08:00",
"approver" : "test_uesr",
"email" : null,
"msg" : null,
"directory_path" : null
} ]
}
}
```

### 状态码：400

#### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

#### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

#### Forbidden

```
{
  "error_code" : "DS.60xx",
```

```
"error_msg" : "The user does not have permission to call this API."  
}
```

**状态码: 404**

Not Found

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class SearchApprovalsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        SearchApprovalsRequest request = new SearchApprovalsRequest();  
        try {  
            SearchApprovalsResponse response = client.searchApprovals(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```



## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchApprovalsRequest()
        response = client.search_approvals(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.SearchApprovalsRequest{}  
response, err := client.SearchApprovals(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是ApprovalVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8.2 撤回审批单

### 功能介绍

撤回审批单。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/approvals

表 8-354 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-355 Query 参数

参数	是否必选	参数类型	描述
ids	是	String	审批单ID数组。

## 请求参数

表 8-356 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-357 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	审批单撤回的返回结果，成功的个数。

表 8-358 data

参数	参数类型	描述
value	String	审批单撤回的对象个数

**状态码：400**

**表 8-359** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-360** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-361** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

**表 8-362** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据审批单ID撤回审批单申请。

```
PUT https://{endpoint}/v2/{project_id}/design/approvals?ids=1230929439261609984
```

## 响应示例

**状态码：200**

Success, 返回data数据是撤回审批条数。

```
{
  "data": {
    "value": 1
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class RollbackApprovalSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        RollbackApprovalRequest request = new RollbackApprovalRequest();
        try {
            RollbackApprovalResponse response = client.rollbackApproval(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = RollbackApprovalRequest()
    response = client.rollback_approval(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RollbackApprovalRequest{}
    response, err := client.RollbackApproval(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是撤回审批条数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8.3 审批单处理

### 功能介绍

审批驳回/通过, 单个或多个action-id=reject/resolve。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/approvals/action

表 8-363 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-364 Query 参数

参数	是否必选	参数类型	描述
action-id	是	String	处理审批单结果类型。 枚举值: <ul style="list-style-type: none"> <li>reject: 审批驳回</li> <li>resolve: 审批通过</li> </ul>



## 请求参数

表 8-365 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-366 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	审批单ID列表，ID字符串。
msg	是	String	审批单信息，审批人填写的审批意见。

## 响应参数

状态码：200

表 8-367 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	审批单撤回的返回结果，成功的个数。

表 8-368 data

参数	参数类型	描述
value	<b>BatchOperationVO</b> object	审批操作返回结果

表 8-369 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID, ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID, ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID, ID字符串。
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值: <ul style="list-style-type: none"> <li>• RUNNING: 运行中</li> <li>• SUCCESS: 操作成功</li> <li>• FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。
groups	Array of <b>BatchOperationVO</b> objects	分组信息。

状态码: 400

表 8-370 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-371 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-372 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-373 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

审批通过物理表发布。

PUT https://{endpoint}/v2/{project\_id}/design/approvals/action?action-id=resolve&from=standard

```
{
  "ids": [ "1214980984198561792" ],
  "msg": "同意! "
}
```

## 响应示例

### 状态码: 200

Success, 返回data数据是执行结果。

```
{
  "data": {
    "value": {
      "id": null,
      "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "group_id": "1230932891752615936",
      "biz_name": null,
      "biz_id": null,
      "operation_status": "SUCCESS",
      "operation_type": "APPROVAL",
      "biz_info": null,
      "create_by": "test_uesr",
      "remark": null,
      "total": 1,
      "success": 1,
      "failed": 0,
      "rate": "1.00",
      "logs": null,
      "groups": [ {
        "id": "1230932895607181312",
        "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "group_id": "1230932891752615936",
        "biz_name": "test_czh_0305_003",
        "biz_id": "1214580364266340352",
        "operation_status": "SUCCESS",
        "operation_type": "APPROVAL",
        "biz_info": "{}",
        "create_by": "test_uesr",
        "remark": "PASSED",
        "total": 0,
        "success": 0,
        "failed": 0,
        "rate": null,
        "logs": null,
        "groups": null
      } ]
    }
  }
}
```

### 状态码: 400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码: 401

Unauthorized

```
{
  "error_code": "DS.60xx",
```

```
"error_msg" : "User authentication failed."  
}
```

**状态码：403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

**状态码：404**

Not Found

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

审批通过物理表发布。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ConfirmApprovalsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ConfirmApprovalsRequest request = new ConfirmApprovalsRequest();  
        ApprovalInfoParam body = new ApprovalInfoParam();  
        List<String> listbodyIds = new ArrayList<>();  
    }  
}
```

```
listbodyIds.add("1214980984198561792");
body.withMsg("同意!");
body.withIds(listbodyIds);
request.withBody(body);
try {
    ConfirmApprovalsResponse response = client.confirmApprovals(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

审批通过物理表发布。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ConfirmApprovalsRequest()
        listbody = [
            "1214980984198561792"
        ]
        request.body = ApprovalInfoParam(
            msg="同意!",
            ids=listbody
        )
        response = client.confirm_approvals(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

审批通过物理表发布。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ConfirmApprovalsRequest{}
    var listIdsbody = []string{
        "1214980984198561792",
    }
    request.Body = &model.ApprovalInfoParam{
        Msg: "同意!",
        Ids: listIdsbody,
    }
    response, err := client.ConfirmApprovals(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是执行结果。
400	BadRequest

状态码	描述
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8.4 批量发布

### 功能介绍

批量发布。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/approvals/batch-publish

表 8-374 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-375 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。



参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

表 8-376 请求 Body 参数

参数	是否必选	参数类型	描述
biz_infos	是	Array of <b>BizInfoVO</b> objects	业务信息列表。
approver_user_id	是	String	审批人ID。
approver_user_name	是	String	审批人姓名。
email	否	String	审批人邮箱, 仅在创建审批人时填写。
fast_approval	否	Boolean	快速审批, 非正式场景, 用于快速上手体验, 仅在当前用户有审批权限时提供。
schedule_time	否	String	作业调度时间。格式参照: 30_18, 表示18点30分。
env_type	否	String	开发生产环境类型。 枚举值: <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>

表 8-377 BizInfoVO

参数	是否必选	参数类型	描述
biz_id	是	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	是	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
env_type	否	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>• INVALID_TYPE: 无效环境</li> <li>• DEV_TYPE: 开发环境</li> <li>• PROD_TYPE: 生产环境</li> <li>• DEV_PROD_TYPE: 开发生产环境</li> </ul>

## 响应参数

状态码：200

表 8-378 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	批量发布的返回结果，成功的个数。

表 8-379 data

参数	参数类型	描述
value	<b>BatchOperationVO</b> object	批量发布操作返回结果

表 8-380 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID，ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID，ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID，ID字符串。
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值： <ul style="list-style-type: none"> <li>● RUNNING: 运行中</li> <li>● SUCCESS: 操作成功</li> <li>● FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。

参数	参数类型	描述
groups	Array of <b>BatchOperationV</b> <b>O</b> objects	分组信息。

状态码：400

表 8-381 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-382 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-383 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-384 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据物理模型ID，批量发布物理模型。

POST https://{endpoint}/v2/{project\_id}/design/approvals/batch-publish?from=standard

```
{
  "biz_infos": [ {
    "biz_id": "1222853173564289024",
    "biz_type": "TABLE_MODEL"
  }, {
    "biz_id": "1229920251379118081",
    "biz_type": "TABLE_MODEL"
  } ],
  "approver_user_id": "0833a573fe80d5401f6dc010a775569a",
  "approver_user_name": "test_uesr",
  "email": null,
  "fast_approval": true,
  "schedule_time": "00_00"
}
```

## 响应示例

状态码：200

Success，返回data数据是 BatchOperationVO对象，批量对象在 BatchOperationVO.group属性中。

```
{
  "data": {
    "value": {
      "id": null,
      "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "group_id": "1230938450061053952",
      "biz_name": null,
      "biz_id": null,
      "operation_status": "SUCCESS",
      "operation_type": "PUBLISH",
      "biz_info": null,
      "create_by": "test_uesr",
      "remark": null,
      "total": 2,
      "success": 2,
      "failed": 0,
      "rate": "1.00",
      "logs": null,
      "groups": [ {
        "id": "1230938456012771328",
        "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "group_id": "1230938450061053952",
        "biz_name": "test_czh_0328",
        "biz_id": "1222853173564289024",
        "operation_status": "SUCCESS",
        "operation_type": "PUBLISH",

```

```
"biz_info" : "{}",
"create_by" : "test_uesr",
"remark" : "PASSED",
"total" : 0,
"success" : 0,
"failed" : 0,
"rate" : null,
"logs" : null,
"groups" : null
}, {
  "id" : "1230938456012771329",
  "tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "group_id" : "1230938450061053952",
  "biz_name" : "test_czh_0416",
  "biz_id" : "1229920251379118081",
  "operation_status" : "SUCCESS",
  "operation_type" : "PUBLISH",
  "biz_info" : "{}",
  "create_by" : "test_uesr",
  "remark" : "PASSED",
  "total" : 0,
  "success" : 0,
  "failed" : 0,
  "rate" : null,
  "logs" : null,
  "groups" : null
}]
}
```

#### 状态码：400

##### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

##### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

##### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

#### 状态码：404

##### Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据物理模型ID，批量发布物理模型。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchPublishSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchPublishRequest request = new BatchPublishRequest();
        ApprovalBatchParam body = new ApprovalBatchParam();
        [](model.BizInfoVo) listbodyBizInfos = new ArrayList<>();
        listbodyBizInfos.add(
            new BizInfoVO()
                .withBizId("1222853173564289024")
                .withBizType(BizInfoVO.BizTypeEnum.fromValue("TABLE_MODEL"))
        );
        listbodyBizInfos.add(
            new BizInfoVO()
                .withBizId("1229920251379118081")
                .withBizType(BizInfoVO.BizTypeEnum.fromValue("TABLE_MODEL"))
        );
        body.withScheduleTime("00_00");
        body.withFastApproval(true);
        body.withApproverUserName("test_uesr");
        body.withApproverUserId("0833a573fe80d5401f6dc010a775569a");
        body.withBizInfos(listbodyBizInfos);
        request.withBody(body);
        try {
            BatchPublishResponse response = client.batchPublish(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```



```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

根据物理模型ID，批量发布物理模型。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchPublishRequest()
        listBizInfosbody = [
            BizInfoVO(
                biz_id="1222853173564289024",
                biz_type="TABLE_MODEL"
            ),
            BizInfoVO(
                biz_id="1229920251379118081",
                biz_type="TABLE_MODEL"
            )
        ]
        request.body = ApprovalBatchParam(
            schedule_time="00_00",
            fast_approval=True,
            approver_user_name="test_uesr",
            approver_user_id="0833a573fe80d5401f6dc010a775569a",
            biz_infos=listBizInfosbody
        )
        response = client.batch_publish(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据物理模型ID，批量发布物理模型。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchPublishRequest{}
    bizTypeBizInfos := model.GetBizInfoVoBizTypeEnum().TABLE_MODEL
    bizTypeBizInfos1 := model.GetBizInfoVoBizTypeEnum().TABLE_MODEL
    var listBizInfosbody = []model.BizInfoVo{
        {
            BizId: "1222853173564289024",
            BizType: &bizTypeBizInfos,
        },
        {
            BizId: "1229920251379118081",
            BizType: &bizTypeBizInfos1,
        },
    }
    scheduleTimeApprovalBatchParam := "00_00"
    fastApprovalApprovalBatchParam := true
    request.Body = &model.ApprovalBatchParam{
        ScheduleTime: &scheduleTimeApprovalBatchParam,
        FastApproval: &fastApprovalApprovalBatchParam,
        ApproverUserName: "test_uesr",
        ApproverUserId: "0833a573fe80d5401f6dc010a775569a",
        BizInfos: listBizInfosbody,
    }
    response, err := client.BatchPublish(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 BatchOperationVO对象，批量对象在 BatchOperationVO.group属性中。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8.5 批量下线

### 功能介绍

批量下线。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/approvals/batch-offline

表 8-385 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-386 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-387 请求 Body 参数

参数	是否必选	参数类型	描述
biz_infos	是	Array of <a href="#">BizInfoVO</a> objects	业务信息列表。
approver_user_id	是	String	审批人ID。
approver_user_name	是	String	审批人姓名。
email	否	String	审批人邮箱，仅在创建审批人时填写。
fast_approval	否	Boolean	快速审批，非正式场景，用于快速上手体验，仅在当前用户有审批权限时提供。
schedule_time	否	String	作业调度时间。格式参照：30_18，表示18点30分。

参数	是否必选	参数类型	描述
env_type	否	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>

表 8-388 BizInfoVO

参数	是否必选	参数类型	描述
biz_id	是	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	是	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
env_type	否	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>• INVALID_TYPE: 无效环境</li> <li>• DEV_TYPE: 开发环境</li> <li>• PROD_TYPE: 生产环境</li> <li>• DEV_PROD_TYPE: 开发生产环境</li> </ul>

## 响应参数

状态码：200

表 8-389 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	批量下线的返回结果，成功的个数。

表 8-390 data

参数	参数类型	描述
value	<b>BatchOperationVO</b> object	批量下线操作返回结果

表 8-391 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID，ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID，ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID，ID字符串。
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值： <ul style="list-style-type: none"> <li>● RUNNING: 运行中</li> <li>● SUCCESS: 操作成功</li> <li>● FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。



参数	参数类型	描述
groups	Array of <b>BatchOperationV</b> <b>O</b> objects	分组信息。

状态码：400

表 8-392 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-393 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-394 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-395 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据物理表ID等参数，批量下线物理表。

```
POST https://{endpoint}/v2/{project_id}/design/approvals/batch-offline?from=standard
{
  "biz_infos": [ {
    "biz_id": "1229920251379118081",
    "biz_type": "TABLE_MODEL"
  }, {
    "biz_id": "1222853173564289024",
    "biz_type": "TABLE_MODEL"
  } ],
  "approver_user_id": "0833a573fe80d5401f6dc010a775569a",
  "approver_user_name": "test_uesr",
  "email": null,
  "fast_approval": true
}
```

## 响应示例

**状态码：200**

Success，返回data数据是 BatchOperationVO对象，批量对象在 BatchOperationVO.group属性中。

```
{
  "data": {
    "value": {
      "id": null,
      "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "group_id": "1230939932055457792",
      "biz_name": null,
      "biz_id": null,
      "operation_status": "SUCCESS",
      "operation_type": "OFFLINE",
      "biz_info": null,
      "create_by": "test_uesr",
      "remark": null,
      "total": 2,
      "success": 2,
      "failed": 0,
      "rate": "1.00",
      "logs": null,
      "groups": [ {
        "id": "1230939939395489792",
        "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "group_id": "1230939932055457792",
        "biz_name": "test_czh_0416",
        "biz_id": "1229920251379118081",
        "operation_status": "SUCCESS",
        "operation_type": "OFFLINE",
        "biz_info": "{}",

```

```
"create_by": "test_uesr",
"remark": "PASSED",
"total": 0,
"success": 0,
"failed": 0,
"rate": null,
"logs": null,
"groups": null
}, {
  "id": "1230939939395489793",
  "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "group_id": "1230939932055457792",
  "biz_name": "test_czh_0328",
  "biz_id": "1222853173564289024",
  "operation_status": "SUCCESS",
  "operation_type": "OFFLINE",
  "biz_info": "{}",
  "create_by": "test_uesr",
  "remark": "PASSED",
  "total": 0,
  "success": 0,
  "failed": 0,
  "rate": null,
  "logs": null,
  "groups": null
}]
}
```

#### 状态码：400

##### BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

#### 状态码：401

##### Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

#### 状态码：403

##### Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

#### 状态码：404

##### Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

根据物理表ID等参数，批量下线物理表。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchOfflineSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchOfflineRequest request = new BatchOfflineRequest();
        ApprovalBatchParam body = new ApprovalBatchParam();
        List<BizInfoVO> listbodyBizInfos = new ArrayList<>();
        listbodyBizInfos.add(
            new BizInfoVO()
                .withBizId("1229920251379118081")
                .withBizType(BizInfoVO.BizTypeEnum.fromValue("TABLE_MODEL"))
        );
        listbodyBizInfos.add(
            new BizInfoVO()
                .withBizId("1222853173564289024")
                .withBizType(BizInfoVO.BizTypeEnum.fromValue("TABLE_MODEL"))
        );
        body.withFastApproval(true);
        body.withApproverUserName("test_uesr");
        body.withApproverUserId("0833a573fe80d5401f6dc010a775569a");
        body.withBizInfos(listbodyBizInfos);
        request.withBody(body);
        try {
            BatchOfflineResponse response = client.batchOffline(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

根据物理表ID等参数，批量下线物理表。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchOfflineRequest()
        listBizInfosbody = [
            BizInfoVO(
                biz_id="1229920251379118081",
                biz_type="TABLE_MODEL"
            ),
            BizInfoVO(
                biz_id="1222853173564289024",
                biz_type="TABLE_MODEL"
            )
        ]
        request.body = ApprovalBatchParam(
            fast_approval=True,
            approver_user_name="test_uesr",
            approver_user_id="0833a573fe80d5401f6dc010a775569a",
            biz_infos=listBizInfosbody
        )
        response = client.batch_offline(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据物理表ID等参数，批量下线物理表。

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchOfflineRequest{}
    bizTypeBizInfos:= model.GetBizInfoVoBizTypeEnum().TABLE_MODEL
    bizTypeBizInfos1:= model.GetBizInfoVoBizTypeEnum().TABLE_MODEL
    var listBizInfosbody = []model.BizInfoVo{
        {
            BizId: "1229920251379118081",
            BizType: &bizTypeBizInfos,
        },
        {
            BizId: "1222853173564289024",
            BizType: &bizTypeBizInfos1,
        },
    }
    fastApprovalApprovalBatchParam:= true
    request.Body = &model.ApprovalBatchParam{
        FastApproval: &fastApprovalApprovalBatchParam,
        ApproverUserName: "test_uesr",
        ApproverUserId: "0833a573fe80d5401f6dc010a775569a",
        BizInfos: listBizInfosbody,
    }
    response, err := client.BatchOffline(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 BatchOperationVO对象, 批量对象在 BatchOperationVO.group属性中。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8.6 创建审批人

### 功能介绍

创建审批人。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/approvals/users

表 8-396 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-397 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 ( 响应消息头中X-Subject-Token的值) 使用Token认证时必选。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-398 请求 Body 参数

参数	是否必选	参数类型	描述
app_name	否	String	调用审核系统的应用名称，开发人员自己定。
approver_name	是	String	审批人姓名，需要被添加到相应的空间中，并且该用户须是管理员、开发者，或者是有审批权限的角色。
user_id	是	String	审批人ID。
email	否	String	邮箱地址。
phone_number	否	String	电话号码。
email_notify	否	Boolean	邮件通知。
sms_notify	否	Boolean	短信通知。

## 响应参数

状态码：200

表 8-399 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。



表 8-400 data

参数	参数类型	描述
value	ApproverVO object	value, 统一的返回结果的外层数据结构。

表 8-401 ApproverVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
approver_name	String	审批人姓名。
user_id	String	审批人ID。
user_name	String	审批人名称。
email	String	email信息。
user_type	String	用户类型。 枚举值: <ul style="list-style-type: none"> <li>BIZ_METRIC_OWNER: 业务指标责任人</li> <li>APPROVER: 审批人</li> <li>BIZ_METRIC_OWNER_AND_APPROVER: 业务指标责任人是审核人</li> </ul>
phone_number	String	电话号码。
create_by	String	创建人。
create_time	String	创建时间, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
app_name	String	业务系统名称。
topic_urn	String	smn主题urn。
project_id	String	项目ID。

状态码: 400

表 8-402 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

#### 状态码：401

表 8-403 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

#### 状态码：403

表 8-404 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

#### 状态码：404

表 8-405 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数创建审批人。

```
POST https://{endpoint}/v2/{project_id}/design/approvals/users?from=standard
{
```



## Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数创建审批人。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateApproverSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateApproverRequest request = new CreateApproverRequest();
        ApproverParam body = new ApproverParam();
        body.withSmsNotify(true);
        body.withEmailNotify(true);
        body.withPhoneNumber("17775558866");
        body.withEmail("88664488@hw.com");
        body.withUserId("0833a573fe80d5401f6dc010a775569a");
        body.withApproverName("test_uesr");
        body.withAppName("dlg-ds");
        request.withBody(body);
        try {
            CreateApproverResponse response = client.createApprover(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

根据请求参数创建审批人。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateApproverRequest()
        request.body = ApproverParam(
            sms_notify=True,
            email_notify=True,
            phone_number="17775558866",
            email="88664488@hw.com",
            user_id="0833a573fe80d5401f6dc010a775569a",
            approver_name="test_uesr",
            app_name="dlg-ds"
        )
        response = client.create_approver(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据请求参数创建审批人。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
```

```

)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateApproverRequest{
        smsNotifyApproverParam:= true
        emailNotifyApproverParam:= true
        phoneNumberApproverParam:= "17775558866"
        emailApproverParam:= "88664488@hw.com"
        appNameApproverParam:= "dlg-ds"
        request.Body = &model.ApproverParam{
            SmsNotify: &smsNotifyApproverParam,
            EmailNotify: &emailNotifyApproverParam,
            PhoneNumber: &phoneNumberApproverParam,
            Email: &emailApproverParam,
            UserId: "0833a573fe80d5401f6dc010a775569a",
            ApproverName: "test_uesr",
            AppName: &appNameApproverParam,
        }
    }
    response, err := client.CreateApprover(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是ApproverVO。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8.7 查询审批人列表

### 功能介绍

查询审批人列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/approvals/users

表 8-406 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-407 Query 参数

参数	是否必选	参数类型	描述
approver_name	否	String	审核人姓名。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-408 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-409 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-410 data

参数	参数类型	描述
value	<a href="#">value</a> object	value，统一的返回结果的外层数据结构。



表 8-411 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>ApproverVO</b> objects	查询到的审批人对象 ( ApproverVO ) 数组。

表 8-412 ApproverVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
approver_name	String	审批人姓名。
user_id	String	审批人ID。
user_name	String	审批人名称。
email	String	email信息。
user_type	String	用户类型。 枚举值: <ul style="list-style-type: none"> <li>• BIZ_METRIC_OWNER: 业务指标责任人</li> <li>• APPROVER: 审批人</li> <li>• BIZ_METRIC_OWNER_AND_APPROVER: 业务指标责任人是审核人</li> </ul>
phone_number	String	电话号码。
create_by	String	创建人。
create_time	String	创建时间, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
app_name	String	业务系统名称。
topic_urn	String	smn主题urn。
project_id	String	项目ID。

状态码: 400

表 8-413 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-414 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-415 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-416 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询审批人列表。

GET https://{endpoint}/v2/{project\_id}/design/approvals/users

## 响应示例

### 状态码：200

Success, 返回data数据是ApproverVO列表和数目。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "42821",
        "approver_name": "test_uesr",
        "user_id": "0833a573fe80d5401f6dc010a775569a",
        "user_name": "test_uesr",
        "email":
"B*****user",
        "user_type": null,
        "phone_number":
"B*****E",
        "create_by": "test_uesr",
        "create_time": "2024-04-19T09:59:58.000Z",
        "app_name": "dlg_ds",
        "topic_urn": "urn:smn:cn-north-7:0833a5737480d53b2f25c010dc1a7b88:DataArts_Topic_Approver_cn-north-7_0833a5737480d53b2f25c010dc1a7b88_dlg_ds_d8wwN1SH",
        "project_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cf08fda4f3b95faa5ba65fa5ec3"
      } ]
    }
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

### 状态码：404

Not Found

```
{
  "error_code": "DS.60xx",
}
```

```
"error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListApproversSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListApproversRequest request = new ListApproversRequest();  
        try {  
            ListApproversResponse response = client.listApprovers(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListApproversRequest()
        response = client.list_approvers(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListApproversRequest{}
    response, err := client.ListApprovers(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```

```

        fmt.Println(err)
    }
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是ApproverVO列表和数目。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8.8 删除审批人

### 功能介绍

删除审批人。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/approvals/users

表 8-417 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-418 Query 参数

参数	是否必选	参数类型	描述
approver_ids	是	String	审批人id

## 请求参数

表 8-419 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-420 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-421 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-422 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-423 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-424 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-425 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据审批人ID删除审批人。



```
DELETE https://{endpoint}/v2/{project_id}/design/approvals/users?approver_ids=38762
```

## 响应示例

### 状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

### 状态码：404

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteApproverSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    }
}
```

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteApproverRequest request = new DeleteApproverRequest();
try {
    DeleteApproverResponse response = client.deleteApprover(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteApproverRequest()
        response = client.delete_approver(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteApproverRequest{}
    response, err := client.DeleteApprover(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是删除数目。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.8.9 删除实体下展

### 功能介绍

当已发布的实体被编辑时，其会生成下展，该接口用于删除实体的下展信息。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/approvals/business/{biz\_id}

表 8-426 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
biz_id	是	String	待删除下展的实体ID，ID字符串。

表 8-427 Query 参数

参数	是否必选	参数类型	描述
biz_type	是	String	待删除下展的实体类型。 枚举值： <ul style="list-style-type: none"> <li>• ATOMIC_INDEX: 原子指标</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DIMENSION: 维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• TABLE_MODEL: 关系建模：逻辑实体/物理表</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• CODE_TABLE: 码表</li> <li>• BIZ_METRIC: 业务指标</li> <li>• COMPOUND_METRIC: 复合指标</li> </ul>

## 请求参数

表 8-428 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-429 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-430 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-431 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-432 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-433 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

删除关系建模中表ID为1217123755210469376的下展信息。

```
DELETE https://{endpoint}/v2/{project_id}/design/approvals/business/1217123755210469376?
biz_type=TABLE_MODEL
```

## 响应示例

状态码：200

Success，返回data数据是删除个数，为1则删除成功。

```
{
  "data": {
    "value": 1
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteDesignLatestApprovalSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteDesignLatestApprovalRequest request = new DeleteDesignLatestApprovalRequest();
        request.withBizId("{biz_id}");
```

```
try {
    DeleteDesignLatestApprovalResponse response = client.deleteDesignLatestApproval(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteDesignLatestApprovalRequest()
        request.biz_id = "{biz_id}"
        response = client.delete_design_latest_approval(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```



```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteDesignLatestApprovalRequest{
    request.BizId = "{biz_id}"
}
response, err := client.DeleteDesignLatestApproval(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数，为1则删除成功。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.8.10 获取下展信息与已发布实体的差异

### 功能介绍

当已发布的实体被编辑时，其会生成下展，该接口用于获取下展信息与已发布实体的差异。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/design/approvals/business/{biz\_id}/diff

表 8-434 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
biz_id	是	String	待比较下展的实体ID，ID字符串。

表 8-435 Query 参数

参数	是否必选	参数类型	描述
biz_type	是	String	待删除下展的实体类型。 枚举值： <ul style="list-style-type: none"> <li>• ATOMIC_INDEX: 原子指标</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DIMENSION: 维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• TABLE_MODEL: 关系建模：逻辑实体/物理表</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• CODE_TABLE: 码表</li> <li>• BIZ_METRIC: 业务指标</li> <li>• COMPOUND_METRIC: 复合指标</li> </ul>

## 请求参数

表 8-436 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-437 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	接口返回的数据。

表 8-438 data

参数	参数类型	描述
value	<a href="#">PublishVersionVO</a> object	版本信息。

表 8-439 PublishVersionVO

参数	参数类型	描述
id	String	版本ID，ID字符串。

参数	参数类型	描述
version_name	String	版本名称。
version_tag	String	版本标记，只读。
description	String	版本描述。
biz_id	String	业务对象ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	业务详情，只读。
biz_info_vo	Object	业务对象。
effect_objs	String	影响信息，只读。
change_props	String	变化信息，只读。
sql_ddl	String	SQL脚本，只读。
physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
biz_metric	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
is_current_version	Boolean	是否为当前版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人，只读。

**状态码：400**

**表 8-440** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-441** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-442 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

获取关系建模中表ID为1217123755210469376的已发布实体与下展信息之间的差异信息。

```
GET https://{endpoint}/v2/{project_id}/design/approvals/business/1217123755210469376/diff?
biz_type=TABLE_MODEL
```

## 响应示例

状态码：200

Success，返回data数据是PublishVersionVO数组，以及总数。

```
{
  "data" : {
    "value" : {
      "id" : "1227929420048830464",
      "version_name" : "test_l2_2_2024-04-11_10-33-07_576",
      "version_tag" : "test_l2_2_2024-04-11_10-33-07_576",
      "description" : null,
      "biz_id" : "1217123755210469376",
      "biz_type" : "TABLE_MODEL",
      "biz_info" : {"versionName":null,"versionTag":null,"versionId":null,"createBy
      \": \"0833a573fe80d5401f6dc010a775569a\", \"updateBy\": \"0833a573fe80d5401f6dc010a775569a
      \", \"createTime\": 1710226516000, \"updateTime\": 1712802684801, \"l3Id\": \"1123655694424358912\", \"l2Id
      \": \"1123655694424358912\", \"alias\": null, \"encoding\": \"test_l2_2\", \"prefix\": \"test_l2_\", \"codeNum
      \": null, \"extendInfo\": {\"columnTag\": {\"columnSecrecyLevel\": {\"tableTag\": {\"dirtyOutSwitch\": {\"dirtyOutDatabase\": {\"dirtyOutPrefix\": {\"err_
      \": {\"dirtyOutSuffix\": {\"id\": \"1217123755210469376\", \"description\": \"无\", \"tenantId
      \": \"0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70\", \"status
      \": \"DRAFT\", \"tbName\": \"test_l2_2\", \"tbLogicName\": \"test_l2_2\", \"owner\": \"\", \"queueName
      \": null, \"schema\": null, \"dbName\": null, \"distribute\": null, \"distributeColumn\": null, \"compression
      \": null, \"obsLocation\": null, \"preCombineField\": null, \"dwd
      \": \"5013aa3f86df4335945131ce9d242b80\", \"dwName\": \"RDS-Mysql\", \"dwType\": \"MYSQL
      \", \"physicalTable\": \"CREATE_FAILED\", \"devPhysicalTable\": \"NO_NEED\", \"technicalAsset
      \": \"CREATE_FAILED\", \"businessAsset\": \"NO_NEED\", \"metaDataLink\": \"UPDATE_FAILED\", \"dataQuality
      \": \"NO_NEED\", \"dlfTask\": \"NO_NEED\", \"summaryStatus\": \"SUMMARY_FAILED\", \"materialization
      \": null, \"publishToDlm\": null, \"envType\": \"PROD_TYPE\", \"devVersion\": null, \"prodVersion\": null, \"apiId
      \": null, \"tbLogicGuid\": \"30a94acc-d5bd-44cf-a751-6fb82989a7c2\", \"tbGuid\": null, \"tblId
      \": null, \"partitionConf\": null, \"qualityOwner\": null, \"qualityId\": null, \"useRecentlyPartition\": false, \"reversed
      \": null, \"dirtyOutSwitch\": true, \"dirtyOutDatabase\": \"default\", \"dirtyOutPrefix\": \"err_\", \"dirtyOutSuffix
      \": \"\", \"dlfTaskId\": null, \"selfDefinedFields\": {\"fdNameCh\": \"aaa\", \"fdNameEn\": \"aaa\", \"notNull
      \": false, \"fdValue\": \"\", \"id\": null, \"status\": null, \"fdName\": \"aaa\", \"tenantId\": null, \"nameCh
      \": null, \"nameEn\": null, \"createBy\": null, \"l3Id\": null}, \"scheduleTime\": null, \"modelId
      \": \"1217123720355803136\", \"parentTableId\": null, \"parentTableName\": null, \"parentTableCode
      \": null, \"relatedLogicTableId\": \"1210601123904405504\", \"relatedLogicTableName
      \": \"test_l2_2\", \"relatedLogicTableModelId\": \"1184439352260976640\", \"relatedLogicTableModelName
      \": \"wgttest\", \"logicTbName\": \"test_l2_2\", \"logicTbGuid\": \"30a94acc-d5bd-44cf-
      a751-6fb82989a7c2\", \"logicTbId\": null, \"bizCatalogId\": \"1123655694424358912\", \"bizCatalogGuid
      \": null, \"catalogPath\": {\"l1Id\": \"1088848987810824192\", \"l2Id\": \"1123655694424358912\"}}
```

```
"/"l3Id"/"l4Id"/"l5Id"/"l6Id"/"l7Id"/"secretType":"PUBLIC","tableType":"MYSQL_TABLE","dataFormat":"Parquet","configs":{"obsBucket":null,"approval":null,"tags":[{"versionName":null,"versionTag":null,"versionId":null,"createBy":"test_uesr","updateBy":"0833a573fe80d5401f6dc010a775569a","createTime":1712800606000,"updateTime":1712801851000,"l3Id":null,"l2Id":null,"alias":null,"encoding":null,"prefix":null,"codeNum":null,"id":"1196773541052301312","tagId":"1196773541052301312","bizId":"1217123755210469376","tagName":"aaa","bizType":"TABLE_MODEL","tenantId":"0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70","status":null,"nameCh":null,"nameEn":null,"lastL2Id":null},"secrecyLevels":null,"attributes":[{"versionName":null,"versionTag":null,"versionId":null,"createBy":"0833a573fe80d5401f6dc010a775569a","updateBy":"0833a573fe80d5401f6dc010a775569a","createTime":1710226516000,"updateTime":1710849845000,"l3Id":null,"l2Id":null,"alias":null,"encoding":"LP070248","prefix":"LP","codeNum":"070248","id":"1217123755210469376","nameEn":"","id":"","description":null,"obsLocation":null,"dataType":"VARCHAR","isPrimaryKey":true,"isPartitionKey":false,"isForeignKey":false,"extendField":false,"ordinal":1,"tableModelId":null,"relatedLogicAttrId":"1210601124399333376","relatedLogicAttrName":"id","relatedLogicAttrNameEn":"id","standRowId":null,"standRowName":null,"tags":[{"versionName":null,"versionTag":null,"versionId":null,"createBy":"test_uesr","updateBy":"test_uesr","createTime":1712802684000,"updateTime":1712802684000,"l3Id":null,"l2Id":null,"alias":null,"encoding":null,"prefix":null,"codeNum":null,"id":"112667","tagId":"1196773541052301312","bizId":"1217123755210469376","tagName":"aaa","bizType":"TABLE_MODEL","tenantId":"0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70","status":null,"nameCh":null,"nameEn":null,"lastL2Id":null},"secrecyLevels":null,"dataTypeExtend":null,"domainType":"STRING","qualityInfos":[],"standElementValues":null,"selfDefinedFields":[],"notNull":true,"code":"LP070248","bizCatalogId":null,"status":null,"tenantId":null,"lastL2Id":null},"versionName":null,"versionTag":null,"versionId":null,"createBy":null,"updateBy":null,"createTime":1712802684756,"updateTime":1712802684756,"l3Id":null,"l2Id":null,"alias":"","encoding":"","prefix":null,"codeNum":null,"id":"0","nameEn":"add_column","nameCh":"add_column","description":null,"obsLocation":null,"dataType":"DOUBLE","isPrimaryKey":false,"isPartitionKey":false,"isForeignKey":false,"extendField":false,"ordinal":2,"tableModelId":null,"relatedLogicAttrId":null,"relatedLogicAttrName":null,"relatedLogicAttrNameEn":null,"standRowId":null,"standRowName":null,"tags":[],"secrecyLevels":null,"dataTypeExtend":"","domainType":"NUMBER","qualityInfos":null,"standElementValues":null,"selfDefinedFields":[],"notNull":false,"code":"","bizCatalogId":null,"status":null,"tenantId":null,"lastL2Id":null},"relations":[{"versionName":null,"versionTag":null,"versionId":null,"createBy":"test_uesr","updateBy":"test_uesr","createTime":1712802684757,"updateTime":1712802684757,"l3Id":null,"l2Id":null,"alias":null,"encoding":null,"prefix":null,"codeNum":null,"id":"1219738211841208320","name":"test_l2_1_test_l2_2_1-wg测试","role":null,"tenantId":"0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70","sourceTableId":"1217123755206275075","sourceTableName":"test_l2_1","sourceTableDb":null,"targetTableId":"1217123755210469376","targetTableName":"test_l2_2","targetTableDb":null,"sourceType":"ONE","targetType":"ONE","mappings":[{"versionName":null,"versionTag":null,"versionId":null,"createBy":"test_uesr","updateBy":"test_uesr","createTime":1712802684757,"updateTime":1712802684757,"l3Id":null,"l2Id":null,"alias":null,"encoding":null,"prefix":null,"codeNum":null,"id":"1219738214261321728","relationId":"1219738211841208320","sourceFieldId":"1217123755206275076","sourceFieldName":"id","targetFieldId":"1217123755210469376","targetFieldName":"id","status":null,"nameCh":null,"nameEn":null,"lastL2Id":null},"status":null,"nameCh":"test_l2_1_test_l2_2_1-wg测试","nameEn":"test_l2_1_test_l2_2_1-wg测试","lastL2Id":null},"reverseRelations":[],"mappings":null,"isPartition":false,"code":"test_l2_2","nameCh":"test_l2_2","nameEn":"test_l2_2","tagList":[{"versionName":null,"versionTag":null,"versionId":null,"createBy":null,"updateBy":null,"createTime":1712802787575,"updateTime":1712802787575,"l3Id":null,"l2Id":null,"alias":null,"encoding":null,"prefix":null,"codeNum":null,"id":"1196773541052301312","name":"aaa","tenantId":null,"description":null,"newBiz":null,"status":null,"nameCh":null,"nameEn":null,"lastL2Id":null},"secrecyLevelList":[],"lastL2Id":"1123655694424358912"},"biz_info_vo": {"id": "1217123755210469376", "model_id": "1217123720355803136", "parent_table_id": null, "parent_table_name": null, "parent_table_code": null, "related_logic_table_id": "1210601123904405504", "related_logic_table_name": "test_l2_2", "related_logic_table_model_id": "1184439352260976640", "related_logic_table_model_name": "wgtest", "model": {"id": "1217123720355803136", "name": "mysql",
```

```

"description": "",
"is_physical": true,
"frequent": false,
"top": true,
"level": "SDI",
"dw_type": "MYSQL",
"create_time": 1710226508,
"update_time": 1710991309,
"create_by": "test_uesr",
"update_by": "test_uesr",
"type": "THIRD_NF",
"biz_catalog_ids": null,
"databases": null,
"table_model_prefix": ""
},
"data_format": "Parquet",
"obs_bucket": null,
"obs_location": null,
"configs": "{}",
"table_type": "MYSQL_TABLE",
"owner": "",
"tb_name": "test_l2_2",
"dw_id": "5013aa3f86df4335945131ce9d242b80",
"db_name": null,
"queue_name": null,
"schema": null,
"extend_info": "{\"columnTag\":\"\", \"columnSecrecyLevel\":\"\", \"tableTag\":\"\", \"dirtyOutSwitch\": \"true\", \"dirtyOutDatabase\":\"default\", \"dirtyOutPrefix\":\"err_\", \"dirtyOutSuffix\":\"\"}",
"tb_guid": null,
"tb_id": null,
"logic_tb_name": "test_l2_2",
"logic_tb_guid": "30a94acc-d5bd-44cf-a751-6fb82989a7c2",
"description": "无",
"status": "DRAFT",
"logic_tb_id": null,
"biz_catalog_id": "1123655694424358912",
"catalog_path": "l52/test_l2_3",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": 1710226516,
"update_time": 1.712802684801E9,
"tags": [ {
  "id": "1196773541052301312",
  "tag_id": "1196773541052301312",
  "tag_name": "aaa",
  "biz_id": "1217123755210469376",
  "biz_type": "TABLE_MODEL",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": 1712800606,
  "update_time": 1712801851
} ],
"attributes": [ {
  "id": "1217123755210469377",
  "name_en": "id",
  "name_ch": "id",
  "description": null,
  "obs_location": null,
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "data_type": "VARCHAR",
  "domain_type": "STRING",
  "data_type_extend": null,
  "is_primary_key": true,
  "is_partition_key": false,
  "is_foreign_key": false,
  "extend_field": false,
  "not_null": true,
  "ordinal": 1,

```

```

"table_model_id" : null,
"create_time" : 1710226516,
"update_time" : 1710849845,
"tags" : [ {
  "id" : "112667",
  "tag_id" : "1196773541052301312",
  "tag_name" : "aaa",
  "biz_id" : "1217123755210469377",
  "biz_type" : "TABLE_MODEL",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : 1712802684,
  "update_time" : 1712802684
} ],
"secrecy_levels" : null,
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : [ ],
"alias" : null,
"self_defined_fields" : [ ],
"code" : "LP070248",
"related_logic_attr_id" : "1210601124399333376",
"related_logic_attr_name" : "id",
"related_logic_attr_name_en" : "id"
}, {
  "id" : "0",
  "name_en" : "add_column",
  "name_ch" : "add_column",
  "description" : "",
  "obs_location" : null,
  "create_by" : null,
  "update_by" : null,
  "data_type" : "DOUBLE",
  "domain_type" : "NUMBER",
  "data_type_extend" : "",
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : false,
  "extend_field" : false,
  "not_null" : false,
  "ordinal" : 2,
  "table_model_id" : null,
  "create_time" : 1.712802684756E9,
  "update_time" : 1.712802684756E9,
  "tags" : [ ],
  "secrecy_levels" : null,
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ],
  "code" : "",
  "related_logic_attr_id" : null,
  "related_logic_attr_name" : null,
  "related_logic_attr_name_en" : null
} ],
"mappings" : null,
"relations" : [ {
  "id" : "1219738211841208320",
  "source_table_id" : "1217123755206275075",
  "target_table_id" : "1217123755210469376",
  "name" : "test_l2_1_test_l2_2_1-wg测试",
  "source_table_name" : "test_l2_1",
  "target_table_name" : "test_l2_2",
  "role" : null,
  "tenant_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-
eeb055e69c624311b6b9cfee89a4ec70",
  "source_type" : "ONE",
  "target_type" : "ONE",

```

```
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": 1.712802684757E9,
"update_time": 1.712802684757E9,
"mappings": [ {
  "id": "1219738214261321728",
  "relation_id": "1219738211841208320",
  "source_field_id": "1217123755206275076",
  "target_field_id": "1217123755210469377",
  "source_field_name": "id",
  "target_field_name": "id",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "create_time": 1.712802684757E9,
  "update_time": 1.712802684757E9
} ]
}],
"dw_type": "MYSQL",
"dw_name": "RDS-Mysql",
"l1": "l52",
"l2": "test_l2_3",
"l3": null,
"l1_id": "1088848987810824192",
"l2_id": "1123655694424358912",
"l3_id": null,
"partition_conf": null,
"dfl_task_id": null,
"use_recently_partition": false,
"reversed": null,
"dirty_out_switch": true,
"dirty_out_database": "default",
"dirty_out_prefix": "err_",
"dirty_out_suffix": "",
"quality_owner": null,
"quality_id": null,
"distribute": null,
"distribute_column": null,
"compression": null,
"pre_combine_field": null,
"is_partition": false,
"physical_table": "CREATE_FAILED",
"dev_physical_table": "NO_NEED",
"technical_asset": "CREATE_FAILED",
"business_asset": "NO_NEED",
"meta_data_link": "UPDATE_FAILED",
"data_quality": "NO_NEED",
"summary_status": "SUMMARY_FAILED",
"dev_version": null,
"prod_version": null,
"dev_version_name": null,
"prod_version_name": null,
"env_type": "PROD_TYPE",
"alias": null,
"self_defined_fields": [ {
  "fd_name_ch": "aaa",
  "fd_name_en": "aaa",
  "not_null": false,
  "fd_value": ""
} ],
"code": "test_l2_2",
"has_related_physical_table": false,
"has_related_logic_table": false
},
"effect_objs": null,
"change_props": "基本信息:\r\n数据格式:CCParquet@CC\r\n\r\n\r\n\r\n\r\n表字段:\r\n\r\n\r\nnadd_column : 英文名称:CCadd_column@CC 名称:CCadd_column@CC 数据类型:CCDOUBLE@CC 主键:CCfalse@CC 分区:CCfalse@CC 外键:CCfalse@CC 序号:CC2@CC 数据类型扩展:CC@CC 不为空:CCfalse@CC 编码:CC@CC .\r\n\r\n\r\n关系:\r\n\r\n\r\ntest_l2_1_test_l2_2_1-wg测试 : 名称:CCtest_l2_1_test_l2_2_1-wg测试@CC 子表:CCtest_l2_1@CC 父表:CCtest_l2_2@CC 子对父:CCONE@CC 父对子:CCONE@CC .\r\n\r\n字段映射:\r\n\r\n子表字
```

```
段:CCid@CC 父表字段:CCid@CC .",
  "sql_ddl": "ALTER TABLE ${database}.test_l2_2 ADD add_column DOUBLE;",
  "physical_table": null,
  "technical_asset": null,
  "business_asset": null,
  "meta_data_link": null,
  "data_quality": null,
  "dlf_task": null,
  "materialization": null,
  "publish_to_dlm": null,
  "biz_metric": null,
  "summary_status": null,
  "is_current_version": false,
  "create_time": "2024-04-11T10:33:07.565+08:00",
  "create_by": "test_uesr"
}
}
```

**状态码: 400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码: 401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码: 403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchDesignLatestApprovalDiffSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```



security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
SearchDesignLatestApprovalDiffRequest request = new SearchDesignLatestApprovalDiffRequest();
request.withBizId("{biz_id}");
try {
    SearchDesignLatestApprovalDiffResponse response =
client.searchDesignLatestApprovalDiff(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchDesignLatestApprovalDiffRequest()
        request.biz_id = "{biz_id}"
        response = client.search_design_latest_approval_diff(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchDesignLatestApprovalDiffRequest{}
    request.BizId = "{biz_id}"
    response, err := client.SearchDesignLatestApprovalDiff(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是PublishVersionVO数组，以及总数。
400	BadRequest
401	Unauthorized

状态码	描述
403	Forbidden

## 8.9 主题管理接口

### 8.9.1 查找主题列表

#### 功能介绍

通过名称（支持模糊查询）、创建者、责任人、状态、修改时间分页查找主题。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/subjects

表 8-443 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-444 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
owner	否	String	按负责人查询。

参数	是否必选	参数类型	描述
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。
parent_id	否	String	父目录ID，根节点没有此ID，空值为所有，-1为根节点下节点。ID字符串。

## 请求参数

表 8-445 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-446 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-447 data

参数	参数类型	描述
value	<a href="#">value</a> object	value，统一的返回结果的外层数据结构。

表 8-448 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>CatalogVO</b> objects	查询到的审批单对象 (CatalogVO) 数组。

表 8-449 CatalogVO

参数	参数类型	描述
name_ch	String	中文名称。
name_en	String	英文名称。
description	String	描述信息。
qualified_name	String	扩展名。
guid	String	guid, 自动生成。
code	String	编码。
alias	String	别名。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
data_owner	String	数据所有者。
data_owner_list	String	数据所有者集合。
data_department	String	数据域。
path	String	路径信息。
level	Integer	层级信息。
ordinal	Integer	序号。
owner	String	责任人。

参数	参数类型	描述
parent_id	String	父目录ID，没有则为根目录，ID字符串。
swap_order_id	String	同层排序，目标节点的ID，ID字符串。
id	String	主题ID，ID字符串。
qualified_id	String	认证ID，自动生成。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
children_num	Integer	拥有子流程的数量，不包括子流程的子流程，前端不传。
children	Array of <b>CatalogVO</b> objects	下层子目录，只读。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	属性自定义项。

表 8-450 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-451 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-452 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-453 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-454 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-455 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询状态为草稿、名称中包含“test\_”的主题列表。

```
GET https://{endpoint}/v2/{project_id}/design/subjects?name=test_&parent_id=-1&status=DRAFT
```

## 响应示例

**状态码：200**

Success，返回data数据是CatalogVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "name_ch": "test_001",
        "name_en": "BO000000",
        "description": null,
        "qualified_name": "BO000000@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "guid": null,
        "code": null,
        "status": "DRAFT",
        "new_biz": null,
        "alias": null,
        "data_owner": "1",
        "data_owner_list": "[\"test_uesr\"]",
        "data_department": null,
        "path": "test_001",
        "level": 1,
        "ordinal": 1,
        "owner": "test_uesr",
        "parent_id": null,
        "swap_order_id": null,
        "id": "1193139703919030272",
        "qualified_id": null,
        "from_public": false,
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "create_time": "2024-01-06T10:31:13+08:00",
        "update_time": "2024-01-06T10:42:30+08:00",
        "self_defined_fields": null,
        "children_num": null,
        "children": null
      } ]
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchSubjectSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
SearchSubjectRequest request = new SearchSubjectRequest();
try {
    SearchSubjectResponse response = client.searchSubject(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchSubjectRequest()
        response = client.search_subject(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchSubjectRequest{}
    response, err := client.SearchSubject(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是CatalogVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.9.2 删除主题

### 功能介绍

删除主题。

### 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v2/{project\_id}/design/subjects

表 8-456 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-457 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-458 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。

参数	是否必选	参数类型	描述
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-459 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-460 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-461 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-462 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。



参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-463** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

**表 8-464** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据ID删除主题信息。

```
DELETE https://{endpoint}/v2/{project_id}/design/subjects
```

## 响应示例

**状态码：200**

Success, 返回data数据是删除个数。

```
{
  "value": 2
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
```

```
"error_msg" : "The user request is illegal."  
}
```

**状态码: 401**

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

**状态码: 403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

**状态码: 404**

Not Found

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class DeleteSubjectSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        DeleteSubjectRequest request = new DeleteSubjectRequest();
        IdsParam body = new IdsParam();
        request.withBody(body);
        try {
            DeleteSubjectResponse response = client.deleteSubject(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSubjectRequest()
        request.body = IdsParam(
        )
        response = client.delete_subject(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
```

```

    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSubjectRequest{}
    request.Body = &model.IdsParam{
    }
    response, err := client.DeleteSubject(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.9.3 创建主题

### 功能介绍

创建主题。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/design/subjects

表 8-465 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-466 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-467 请求 Body 参数

参数	是否必选	参数类型	描述
entity	是	<a href="#">CatalogEntityVO</a> object	catalog实体。

表 8-468 CatalogEntityVO

参数	是否必选	参数类型	描述
typeName	否	String	类型名称，填写“BusinessCatalog”即可（业务分层）。
attributes	是	<b>CatalogAttributeVO</b> object	属性列表。

表 8-469 CatalogAttributeVO

参数	是否必选	参数类型	描述
path	是	String	路径。用“/”作为分隔，如：运营领域/云学院/云学院培训方案。
qualifiedName	是	String	资产名称。
level	是	String	主题所属层级。
name	是	String	名称。
nameEng	是	String	英文名称。
alias	否	String	别名。
description	是	String	描述。
dataOwner	是	String	数据主体。
owner	否	String	责任人。
dataOwnerList	是	Array of strings	数据主体列表。
createTime	否	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
createBy	否	String	创建人。
updateTime	否	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
updateBy	否	String	更新人。
parent	否	<b>parent</b> object	父节点。
parentId	否	String	父节点ID。

参数	是否必选	参数类型	描述
l1	否	Boolean	是否为L1层。主题域分组。
l2	否	Boolean	是否为L2层。主题域。
l3	否	Boolean	是否为L3层。业务对象。
ordinal	否	Integer	顺序编号。主题返回时根据此编号由小到大排序。
tenantId	否	String	租户ID。获取方式参考此接口的路径参数“project_id”获取。
self_defined_fields	否	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。主题的自定义属性项。

表 8-470 parent

参数	是否必选	参数类型	描述
id	否	String	父节点ID。

表 8-471 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

## 响应参数

状态码：200

表 8-472 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-473 data

参数	参数类型	描述
value	<b>CatalogVO</b> object	value, 统一的返回结果的外层数据结构。

表 8-474 CatalogVO

参数	参数类型	描述
name_ch	String	中文名称。
name_en	String	英文名称。
description	String	描述信息。
qualified_name	String	扩展名。
guid	String	guid, 自动生成。
code	String	编码。
alias	String	别名。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理, 只读。
data_owner	String	数据所有者。
data_owner_list	String	数据所有者集合。
data_department	String	数据域。
path	String	路径信息。
level	Integer	层级信息。
ordinal	Integer	序号。
owner	String	责任人。
parent_id	String	父目录ID, 没有则为根目录, ID字符串。



参数	参数类型	描述
swap_order_id	String	同层排序, 目标节点的ID, ID字符串。
id	String	主题ID, ID字符串。
qualified_id	String	认证ID, 自动生成。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
children_num	Integer	拥有子流程的数量, 不包括子流程的子流程, 前端不传。
children	Array of <b>CatalogVO</b> objects	下层子目录, 只读。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	属性自定义项。

表 8-475 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-476 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-477 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-478 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-479 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-480 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数创建主题。

POST https://{endpoint}/v2/{project\_id}/design/subjects

```
{
  "entity": {
    "attributes": {
      "path": "test_create_catalog",
      "qualifiedName": "test_create_catalog@Business",
      "level": "L1",
      "name": "test_create_catalog",
      "nameEng": "test_create_catalog",
      "alias": "",
      "description": "this is a demo.",
      "createTime": 1713576437311,
      "updateTime": 1713576437311,
      "dataOwner": "dataOwner",
      "dataOwnerList": [ "test_uesr" ],
      "self_defined_fields": [ ]
    },
    "typeName": "BusinessCatalog"
  }
}
```

## 响应示例

**状态码：200**

Success，返回data数据是entity详情。

```
{
  "data": {
    "value": {
      "name_ch": "test_create_catalog",
      "name_en": "test_create_catalog",
      "description": "this is a demo.",
      "qualified_name": "test_create_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "guid": null,
      "code": null,
      "status": "DRAFT",
      "new_biz": null,
      "alias": "",
      "data_owner": "dataOwner",
      "data_owner_list": [ "test_uesr" ],
      "data_department": null,
      "path": "test_create_catalog",
      "level": 1,
      "ordinal": 10,
      "owner": "test_uesr",
      "parent_id": null,
      "swap_order_id": null,
    }
  }
}
```

```
"id" : "1231174343611035648",
"qualified_id" : null,
"from_public" : null,
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-04-20T09:27:17.63+08:00",
"update_time" : "2024-04-20T09:27:17.63+08:00",
"self_defined_fields" : [ ],
"children_num" : null,
"children" : null
}
}
```

#### 状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

#### 状态码：404

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数创建主题。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateSubjectSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateSubjectRequest request = new CreateSubjectRequest();
        CatalogParamsVO body = new CatalogParamsVO();
        List<String> listAttributesDataOwnerList = new ArrayList<>();
        listAttributesDataOwnerList.add("test_uesr");
        CatalogAttributeVO attributesEntity = new CatalogAttributeVO();
        attributesEntity.withPath("test_create_catalog")
            .withQualifiedName("test_create_catalog@Business")
            .withLevel("L1")
            .withName("test_create_catalog")
            .withNameEng("test_create_catalog")
            .withAlias("")
            .withDescription("this is a demo.")
            .withDataOwner("dataOwner")
            .withDataOwnerList(listAttributesDataOwnerList)
            .withCreateTime("1713576437311")
            .withUpdateTime("1713576437311");
        CatalogEntityVO entitybody = new CatalogEntityVO();
        entitybody.withTypeName("BusinessCatalog")
            .withAttributes(attributesEntity);
        body.withEntity(entitybody);
        request.withBody(body);
        try {
            CreateSubjectResponse response = client.createSubject(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据请求参数创建主题。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSubjectRequest()
        listDataOwnerListAttributes = [
            "test_uesr"
        ]
        attributesEntity = CatalogAttributeVO(
            path="test_create_catalog",
            qualified_name="test_create_catalog@Business",
            level="L1",
            name="test_create_catalog",
            name_eng="test_create_catalog",
            alias="",
            description="this is a demo.",
            data_owner="dataOwner",
            data_owner_list=listDataOwnerListAttributes,
            create_time="1713576437311",
            update_time="1713576437311"
        )
        entitybody = CatalogEntityVO(
            type_name="BusinessCatalog",
            attributes=attributesEntity
        )
        request.body = CatalogParamsVO(
            entity=entitybody
        )
        response = client.create_subject(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据请求参数创建主题。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
```



```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
  )

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSubjectRequest{}
    var listDataOwnerListAttributes = []string{
        "test_uesr",
    }
    aliasAttributes:= ""
    createTimeAttributes:= "1713576437311"
    updateTimeAttributes:= "1713576437311"
    attributesEntity := &model.CatalogAttributeVo{
        Path: "test_create_catalog",
        QualifiedName: "test_create_catalog@Business",
        Level: "L1",
        Name: "test_create_catalog",
        NameEng: "test_create_catalog",
        Alias: &aliasAttributes,
        Description: "this is a demo.",
        DataOwner: "dataOwner",
        DataOwnerList: listDataOwnerListAttributes,
        CreateTime: &createTimeAttributes,
        UpdateTime: &updateTimeAttributes,
    }
    typeNameEntity:= "BusinessCatalog"
    entitybody := &model.CatalogEntityVo{
        TypeName: &typeNameEntity,
        Attributes: attributesEntity,
    }
    request.Body = &model.CatalogParamsVo{
        Entity: entitybody,
    }
    response, err := client.CreateSubject(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是entity详情。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.9.4 修改主题

### 功能介绍

修改主题。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/subjects

表 8-481 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-482 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 ( 响应消息头中X-Subject-Token的值) 使用Token认证时必选。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-483 请求 Body 参数

参数	是否必选	参数类型	描述
entity	是	<a href="#">CatalogEntityVO</a> object	catalog实体。

表 8-484 CatalogEntityVO

参数	是否必选	参数类型	描述
typeName	否	String	类型名称，填写“BusinessCatalog”即可（业务分层）。
attributes	是	<a href="#">CatalogAttributeVO</a> object	属性列表。

表 8-485 CatalogAttributeVO

参数	是否必选	参数类型	描述
path	是	String	路径。用“/”作为分隔，如：运营领域/云学院/云学院培训方案。
qualifiedName	是	String	资产名称。
level	是	String	主题所属层级。
name	是	String	名称。
nameEng	是	String	英文名称。
alias	否	String	别名。

参数	是否必选	参数类型	描述
description	是	String	描述。
dataOwner	是	String	数据主体。
owner	否	String	责任人。
dataOwnerList	是	Array of strings	数据主体列表。
createTime	否	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
createBy	否	String	创建人。
updateTime	否	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
updateBy	否	String	更新人。
parent	否	<b>parent</b> object	父节点。
parentId	否	String	父节点ID。
l1	否	Boolean	是否为L1层。主题域分组。
l2	否	Boolean	是否为L2层。主题域。
l3	否	Boolean	是否为L3层。业务对象。
ordinal	否	Integer	顺序编号。主题返回时根据此编号由小到大排序。
tenantId	否	String	租户ID。获取方式参考此接口的路径参数“project_id”获取。
self_defined_fields	否	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。主题的自定义属性项。

表 8-486 parent

参数	是否必选	参数类型	描述
id	否	String	父节点ID。

表 8-487 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

## 响应参数

状态码：200

表 8-488 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-489 data

参数	参数类型	描述
value	<b>CatalogVO</b> object	value，统一的返回结果的外层数据结构。

表 8-490 CatalogVO

参数	参数类型	描述
name_ch	String	中文名称。
name_en	String	英文名称。
description	String	描述信息。
qualified_name	String	扩展名。
guid	String	guid，自动生成。
code	String	编码。
alias	String	别名。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。
data_owner	String	数据所有者。
data_owner_list	String	数据所有者集合。
data_department	String	数据域。
path	String	路径信息。
level	Integer	层级信息。
ordinal	Integer	序号。
owner	String	责任人。
parent_id	String	父目录ID，没有则为根目录，ID字符串。
swap_order_id	String	同层排序，目标节点的ID，ID字符串。
id	String	主题ID，ID字符串。
qualified_id	String	认证ID，自动生成。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
children_num	Integer	拥有子流程的数量，不包括子流程的子流程，前端不传。
children	Array of <b>CatalogVO</b> objects	下层子目录，只读。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	属性自定义项。

表 8-491 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-492 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-493 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-494 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-495 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-496 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数修改主题。

PUT https://{endpoint}/v2/{project\_id}/design/subjects

```
{
  "entity": {
    "attributes": {
      "path": "test_update_catalog",
      "qualifiedName": "test_create_catalog@Business",
      "level": "L1",
      "name": "test_update_catalog",
      "nameEng": "test_create_catalog",
      "alias": "",
      "description": "this is a update demo.",
      "createTime": 1713578994820,
      "updateTime": 1713578994820,
      "dataOwner": "dataOwner",
      "dataOwnerList": [ "test_uesr" ],
      "self_defined_fields": [ ]
    },
    "typeName": "BusinessCatalog"
  }
}
```

## 响应示例

**状态码：200**

Success，返回data数据是entity详情。

```
{
  "data": {
    "value": {
      "name_ch": "test_update_catalog",
      "name_en": "test_create_catalog",
      "description": "this is a update demo.",
      "qualified_name": "test_create_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "guid": null,
      "code": null,
      "status": "DRAFT",
      "new_biz": null,
      "alias": "",
      "data_owner": "dataOwner",
      "data_owner_list": [ "test_uesr" ],
      "data_department": null,
      "path": "test_update_catalog",
      "level": 1,
      "ordinal": null,
      "owner": "test_uesr",
      "parent_id": null,
      "swap_order_id": null,
    }
  }
}
```

```
"id" : "1231174343611035648",
"qualified_id" : null,
"from_public" : null,
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-04-20T09:27:18+08:00",
"update_time" : "2024-04-20T10:09:55.034+08:00",
"self_defined_fields" : [ ],
"children_num" : null,
"children" : null
}
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数修改主题。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateSubjectSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSubjectRequest request = new UpdateSubjectRequest();
        CatalogParamsVO body = new CatalogParamsVO();
        List<String> listAttributesDataOwnerList = new ArrayList<>();
        listAttributesDataOwnerList.add("test_uesr");
        CatalogAttributeVO attributesEntity = new CatalogAttributeVO();
        attributesEntity.withPath("test_update_catalog")
            .withQualifiedName("test_create_catalog@Business")
            .withLevel("L1")
            .withName("test_update_catalog")
            .withNameEng("test_create_catalog")
            .withAlias("")
            .withDescription("this is a update demo.")
            .withDataOwner("dataOwner")
            .withDataOwnerList(listAttributesDataOwnerList)
            .withCreateTime("1713578994820")
            .withUpdateTime("1713578994820");
        CatalogEntityVO entitybody = new CatalogEntityVO();
        entitybody.withTypeName("BusinessCatalog")
            .withAttributes(attributesEntity);
        body.withEntity(entitybody);
        request.withBody(body);
        try {
            UpdateSubjectResponse response = client.updateSubject(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据请求参数修改主题。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSubjectRequest()
        listDataOwnerListAttributes = [
            "test_uesr"
        ]
        attributesEntity = CatalogAttributeVO(
            path="test_update_catalog",
            qualified_name="test_create_catalog@Business",
            level="L1",
            name="test_update_catalog",
            name_eng="test_create_catalog",
            alias="",
            description="this is a update demo.",
            data_owner="dataOwner",
            data_owner_list=listDataOwnerListAttributes,
            create_time="1713578994820",
            update_time="1713578994820"
        )
        entitybody = CatalogEntityVO(
            type_name="BusinessCatalog",
            attributes=attributesEntity
        )
        request.body = CatalogParamsVO(
            entity=entitybody
        )
        response = client.update_subject(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据请求参数修改主题。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
  )

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSubjectRequest{}
    var listDataOwnerListAttributes = []string{
        "test_uesr",
    }
    aliasAttributes:= ""
    createTimeAttributes:= "1713578994820"
    updateTimeAttributes:= "1713578994820"
    attributesEntity := &model.CatalogAttributeVo{
        Path: "test_update_catalog",
        QualifiedName: "test_create_catalog@Business",
        Level: "L1",
        Name: "test_update_catalog",
        NameEng: "test_create_catalog",
        Alias: &aliasAttributes,
        Description: "this is a update demo.",
        DataOwner: "dataOwner",
        DataOwnerList: listDataOwnerListAttributes,
        CreateTime: &createTimeAttributes,
        UpdateTime: &updateTimeAttributes,
    }
    typeNameEntity:= "BusinessCatalog"
    entitybody := &model.CatalogEntityVo{
        TypeName: &typeNameEntity,
        Attributes: attributesEntity,
    }
    request.Body = &model.CatalogParamsVo{
        Entity: entitybody,
    }
    response, err := client.UpdateSubject(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是entity详情。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.9.5 获取主题树信息

### 功能介绍

获取数据资产主题树信息l1, l2, l3。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/subjects/business

表 8-497 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-498 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数, 即查询Y条数据。默认值50, 取值范围[1,100]。
offset	否	Integer	查询起始坐标, 即跳过X条数据, 仅支持0或limit的整数倍, 不满足则向下取整, 默认值0。



## 请求参数

表 8-499 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-500 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：400

表 8-501 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-502 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-503 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

获取当前工作空间下，数据资产的主题数信息l1、l2、l3。

```
GET https://{endpoint}/v2/{project_id}/design/subjects/business
```

## 响应示例

状态码：200

Success，返回data数据是所有l1、l2、l3实体。

```
{
  "data": {
    "value": [ {
      "id": "1193142949320474624",
      "name": "test_import",
      "qname": "test_import@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "guid": null,
      "ordinal": 1,
      "row": null,
      "catalog": {
        "versionName": null,
        "versionTag": null,
        "versionId": null,
        "createBy": "0833a573fe80d5401f6dc010a775569a",
        "updateBy": "0833a573fe80d5401f6dc010a775569a",
        "createTime": "2024-01-06T02:44:07.000Z",
        "updateTime": "2024-01-26T03:16:24.000Z",
        "l3Id": null,
        "l2Id": null,
        "alias": null,
        "encoding": null,
        "prefix": null,
        "codeNum": null,

```

```

    "id" : "1193142949320474624",
    "nameCh" : "test_import",
    "nameEn" : "test_import",
    "qualifiedName" : "test_import@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : null,
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : "36999ee2-80df-4cad-b5d5-10c50fbeb401",
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "dataOwner" : "",
    "dataOwnerList" : ["test_uesr\"],
    "dataDepartment" : null,
    "path" : "test_import",
    "parentId" : null,
    "level" : 1,
    "ordinal" : 1,
    "status" : "PUBLISHED",
    "approvalInfo" : null,
    "newBiz" : null,
    "fromPublic" : false,
    "swapOrderId" : null,
    "levelStr" : "1",
    "lastL2Id" : null
  },
  "attributes" : {
    "alias" : null,
    "approvalInfo" : null,
    "code" : null,
    "codeNum" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2024-01-06T02:44:07.000Z",
    "dataDepartment" : null,
    "dataOwner" : "",
    "dataOwnerList" : [ "test_uesr" ],
    "description" : null,
    "encoding" : null,
    "fromPublic" : false,
    "guid" : "36999ee2-80df-4cad-b5d5-10c50fbeb401",
    "id" : "1193142949320474624",
    "l1" : true,
    "l2" : false,
    "l2Id" : null,
    "l3" : false,
    "l3Id" : null,
    "lastL2Id" : null,
    "level" : 1,
    "levelStr" : "1",
    "name" : "test_import",
    "nameCh" : "test_import",
    "nameEn" : "test_import",
    "nameEng" : "test_import",
    "newBiz" : null,
    "ordinal" : 1,
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "parentId" : null,
    "path" : "test_import",
    "prefix" : null,
    "qualifiedName" : "test_import@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "status" : "PUBLISHED",
    "swapOrderId" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateTime" : "2024-01-26T03:16:24.000Z",
    "versionId" : null,
    "versionName" : null,
  }
}

```

```

"versionTag" : null
},
"childs" : [ {
  "id" : "1193143478364811264",
  "name" : "test_import_l2",
  "qname" : "test_import.test_import_l2@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid" : null,
  "ordinal" : 1,
  "row" : null,
  "catalog" : {
    "versionName" : null,
    "versionTag" : null,
    "versionId" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2024-01-06T02:46:13.000Z",
    "updateTime" : "2024-01-26T03:17:18.000Z",
    "l3Id" : null,
    "l2Id" : null,
    "alias" : "",
    "encoding" : null,
    "prefix" : null,
    "codeNum" : null,
    "id" : "1193143478364811264",
    "nameCh" : "test_import_l2",
    "nameEn" : "test_import_l2",
    "qualifiedName" : "test_import.test_import_l2@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : "ada",
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : "d6d60e9c-a17c-4acc-b722-3aa4eb86a70b",
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "dataOwner" : "",
    "dataOwnerList" : ["test_uesr"],
    "dataDepartment" : null,
    "path" : "test_import/test_import_l2",
    "parentId" : "1193142949320474624",
    "level" : 2,
    "ordinal" : 1,
    "status" : "PUBLISHED",
    "approvalInfo" : null,
    "newBiz" : null,
    "fromPublic" : false,
    "swapOrderId" : null,
    "levelStr" : "2",
    "lastL2Id" : null
  },
  "attributes" : {
    "alias" : "",
    "approvalInfo" : null,
    "code" : null,
    "codeNum" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2024-01-06T02:46:13.000Z",
    "dataDepartment" : null,
    "dataOwner" : "",
    "dataOwnerList" : [ "test_uesr" ],
    "description" : "ada",
    "encoding" : null,
    "fromPublic" : false,
    "guid" : "d6d60e9c-a17c-4acc-b722-3aa4eb86a70b",
    "id" : "1193143478364811264",
    "l1" : false,
    "l2" : true,
    "l2Id" : null,
    "l3" : false,

```

```

    "l3Id" : null,
    "lastL2Id" : null,
    "level" : 2,
    "levelStr" : "2",
    "name" : "test_import_l2",
    "nameCh" : "test_import_l2",
    "nameEn" : "test_import_l2",
    "nameEng" : "test_import_l2",
    "newBiz" : null,
    "ordinal" : 1,
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "parentId" : "1193142949320474624",
    "path" : "test_import/test_import_l2",
    "prefix" : null,
    "qualifiedName" : "test_import.test_import_l2@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "status" : "PUBLISHED",
    "swapOrderId" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateTime" : "2024-01-26T03:17:18.000Z",
    "versionId" : null,
    "versionName" : null,
    "versionTag" : null
  },
  "childs" : [ {
    "id" : "1200428564969857024",
    "name" : "test_import_l3",
    "qname" :
"test_import.test_import_l2.test_import_l3@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "guid" : null,
    "ordinal" : 1,
    "row" : null,
    "catalog" : {
      "versionName" : null,
      "versionTag" : null,
      "versionId" : null,
      "createBy" : "0833a573fe80d5401f6dc010a775569a",
      "updateBy" : "0833a573fe80d5401f6dc010a775569a",
      "createTime" : "2024-01-26T05:14:33.000Z",
      "updateTime" : "2024-01-26T05:16:35.000Z",
      "l3Id" : null,
      "l2Id" : null,
      "alias" : "",
      "encoding" : null,
      "prefix" : null,
      "codeNum" : null,
      "id" : "1200428564969857024",
      "nameCh" : "test_import_l3",
      "nameEn" : "test_import_l3",
      "qualifiedName" :
"test_import.test_import_l2.test_import_l3@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "description" : "",
      "owner" : "0833a573fe80d5401f6dc010a775569a",
      "guid" : "a7cfe748-ec64-4fe8-ae55-303e9782aa15",
      "code" : null,
      "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "selfDefinedFields" : [ ],
      "dataOwner" : "",
      "dataOwnerList" : "[\\\"test_uesr\\\"]",
      "dataDepartment" : null,
      "path" : "test_import/test_import_l2/test_import_l3",
      "parentId" : "1193143478364811264",
      "level" : 3,
      "ordinal" : 1,

```

```

        "status": "PUBLISHED",
        "approvalInfo": null,
        "newBiz": null,
        "fromPublic": false,
        "swapOrderId": null,
        "levelStr": "3",
        "lastL2Id": null
    },
    "attributes": {
        "alias": "",
        "approvalInfo": null,
        "code": null,
        "codeNum": null,
        "createBy": "0833a573fe80d5401f6dc010a775569a",
        "createTime": "2024-01-26T05:14:33.000Z",
        "dataDepartment": null,
        "dataOwner": "",
        "dataOwnerList": [ "test_uesr" ],
        "description": "",
        "encoding": null,
        "fromPublic": false,
        "guid": "a7cfe748-ec64-4fe8-ae55-303e9782aa15",
        "id": "1200428564969857024",
        "l1": false,
        "l2": true,
        "l2Id": null,
        "l3": false,
        "l3Id": null,
        "lastL2Id": null,
        "level": 3,
        "levelStr": "3",
        "name": "test_import_l3",
        "nameCh": "test_import_l3",
        "nameEn": "test_import_l3",
        "nameEng": "test_import_l3",
        "newBiz": null,
        "ordinal": 1,
        "owner": "0833a573fe80d5401f6dc010a775569a",
        "parentId": "1193143478364811264",
        "path": "test_import/test_import_l2/test_import_l3",
        "prefix": null,
        "qualifiedName":
"test_import.test_import_l2.test_import_l3@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "selfDefinedFields": [ ],
        "status": "PUBLISHED",
        "swapOrderId": null,
        "tenantId": "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "updateBy": "0833a573fe80d5401f6dc010a775569a",
        "updateTime": "2024-01-26T05:16:35.000Z",
        "versionId": null,
        "versionName": null,
        "versionTag": null
    },
    "childs": [ {
        "id": "1226991880613593088",
        "name": "test_import_l4",
        "qname":
"test_import.test_import_l2.test_import_l3.test_import_l4@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "guid": null,
        "ordinal": 1,
        "row": null,
        "catalog": {
            "versionName": null,
            "versionTag": null,
            "versionId": null,
            "createBy": "0833a573fe80d5401f6dc010a775569a",

```

```

"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2024-04-08T12:27:41.000Z",
"updateTime" : "2024-04-08T12:27:50.000Z",
"l3Id" : null,
"l2Id" : null,
"alias" : "",
"encoding" : null,
"prefix" : null,
"codeNum" : null,
"id" : "1226991880613593088",
"nameCh" : "test_import_l4",
"nameEn" : "test_import_l4",
"qualifiedName" :
"test_import.test_import_l2.test_import_l3.test_import_l4@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "",
"owner" : "0833a573fe80d5401f6dc010a775569a",
"guid" : "8b61b0e7-798e-4227-bf5f-d3301048a04d",
"code" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"dataOwner" : "",
"dataOwnerList" : ["chengzhihuadashuaib"],
"dataDepartment" : null,
"path" : "test_import/test_import_l2/test_import_l3/test_import_l4",
"parentId" : "1200428564969857024",
"level" : 4,
"ordinal" : 1,
"status" : "PUBLISHED",
"approvalInfo" : null,
"newBiz" : null,
"fromPublic" : false,
"swapOrderId" : null,
"levelStr" : "4",
"lastL2Id" : null
},
"attributes" : {
"alias" : "",
"approvalInfo" : null,
"code" : null,
"codeNum" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2024-04-08T12:27:41.000Z",
"dataDepartment" : null,
"dataOwner" : "",
"dataOwnerList" : [ "chengzhihuadashuaib" ],
"description" : "",
"encoding" : null,
"fromPublic" : false,
"guid" : "8b61b0e7-798e-4227-bf5f-d3301048a04d",
"id" : "1226991880613593088",
"l1" : false,
"l2" : true,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 4,
"levelStr" : "4",
"name" : "test_import_l4",
"nameCh" : "test_import_l4",
"nameEn" : "test_import_l4",
"nameEng" : "test_import_l4",
"newBiz" : null,
"ordinal" : 1,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : "1200428564969857024",
"path" : "test_import/test_import_l2/test_import_l3/test_import_l4",

```

```

        "prefix" : null,
        "qualifiedName" :
"test_import.test_import_l2.test_import_l3.test_import_l4@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "selfDefinedFields" : [ ],
        "status" : "PUBLISHED",
        "swapOrderId" : null,
        "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "updateBy" : "0833a573fe80d5401f6dc010a775569a",
        "updateTime" : "2024-04-08T12:27:50.000Z",
        "versionId" : null,
        "versionName" : null,
        "versionTag" : null
    },
    "childs" : [ ],
    "count" : 0,
    "status" : "PUBLISHED"
  } ],
  "count" : 1,
  "status" : "PUBLISHED"
}],
"count" : 1,
"status" : "PUBLISHED"
}],
"count" : 1,
"status" : "PUBLISHED"
}, {
  "id" : "1169309252771299328",
  "name" : "城市交通",
  "qname" : "city_traffic@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid" : null,
  "ordinal" : 2,
  "row" : null,
  "catalog" : {
    "versionName" : null,
    "versionTag" : null,
    "versionId" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2023-11-01T08:17:31.000Z",
    "updateTime" : "2023-11-01T08:17:42.000Z",
    "l3Id" : null,
    "l2Id" : null,
    "alias" : "",
    "encoding" : null,
    "prefix" : null,
    "codeNum" : null,
    "id" : "1169309252771299328",
    "nameCh" : "城市交通",
    "nameEn" : "city_traffic",
    "qualifiedName" : "city_traffic@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : "",
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : "66d3fab6-8870-46ea-a8e3-8427c68b6f27",
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "dataOwner" : "",
    "dataOwnerList" : "[\"test_ues\"]",
    "dataDepartment" : null,
    "path" : "城市交通",
    "parentId" : null,
    "level" : 1,
    "ordinal" : 2,
    "status" : "PUBLISHED",
    "approvalInfo" : null,

```



```

    "newBiz" : null,
    "fromPublic" : false,
    "swapOrderId" : null,
    "levelStr" : "1",
    "lastL2Id" : null
  },
  "attributes" : {
    "alias" : "",
    "approvalInfo" : null,
    "code" : null,
    "codeNum" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2023-11-01T08:17:31.000Z",
    "dataDepartment" : null,
    "dataOwner" : "",
    "dataOwnerList" : [ "test_uesr" ],
    "description" : "",
    "encoding" : null,
    "fromPublic" : false,
    "guid" : "66d3fab6-8870-46ea-a8e3-8427c68b6f27",
    "id" : "1169309252771299328",
    "l1" : true,
    "l2" : false,
    "l2Id" : null,
    "l3" : false,
    "l3Id" : null,
    "lastL2Id" : null,
    "level" : 1,
    "levelStr" : "1",
    "name" : "城市交通",
    "nameCh" : "城市交通",
    "nameEn" : "city_traffic",
    "nameEng" : "city_traffic",
    "newBiz" : null,
    "ordinal" : 2,
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "parentId" : null,
    "path" : "城市交通",
    "prefix" : null,
    "qualifiedName" : "city_traffic@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "status" : "PUBLISHED",
    "swapOrderId" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateTime" : "2023-11-01T08:17:42.000Z",
    "versionId" : null,
    "versionName" : null,
    "versionTag" : null
  },
  "childs" : [ {
    "id" : "1169309374670356480",
    "name" : "行程记录",
    "qname" : "city_traffic.stroke_reminder@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "guid" : null,
    "ordinal" : 1,
    "row" : null,
    "catalog" : {
      "versionName" : null,
      "versionTag" : null,
      "versionId" : null,
      "createBy" : "0833a573fe80d5401f6dc010a775569a",
      "updateBy" : "0833a573fe80d5401f6dc010a775569a",
      "createTime" : "2023-11-01T08:18:00.000Z",
      "updateTime" : "2023-11-01T08:18:56.000Z",
      "l3Id" : null,
      "l2Id" : null,

```

```

"alias" : "",
"encoding" : null,
"prefix" : null,
"codeNum" : null,
"id" : "1169309374670356480",
"nameCh" : "行程记录",
"nameEn" : "stroke_reminder",
"qualifiedName" : "city_traffic.stroke_reminder@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "",
"owner" : "0833a573fe80d5401f6dc010a775569a",
"guid" : "38b518d3-fe05-4a6a-a60a-db5cd8b3c5fa",
"code" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"dataOwner" : "",
"dataOwnerList" : ["test_uesr"],
"dataDepartment" : null,
"path" : "城市交通/行程记录",
"parentId" : "1169309252771299328",
"level" : 2,
"ordinal" : 1,
"status" : "PUBLISHED",
"approvalInfo" : null,
"newBiz" : null,
"fromPublic" : false,
"swapOrderId" : null,
"levelStr" : "2",
"lastL2Id" : null
},
"attributes" : {
"alias" : "",
"approvalInfo" : null,
"code" : null,
"codeNum" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-01T08:18:00.000Z",
"dataDepartment" : null,
"dataOwner" : "",
"dataOwnerList" : [ "test_uesr" ],
"description" : "",
"encoding" : null,
"fromPublic" : false,
"guid" : "38b518d3-fe05-4a6a-a60a-db5cd8b3c5fa",
"id" : "1169309374670356480",
"l1" : false,
"l2" : true,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 2,
"levelStr" : "2",
"name" : "行程记录",
"nameCh" : "行程记录",
"nameEn" : "stroke_reminder",
"nameEng" : "stroke_reminder",
"newBiz" : null,
"ordinal" : 1,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : "1169309252771299328",
"path" : "城市交通/行程记录",
"prefix" : null,
"qualifiedName" : "city_traffic.stroke_reminder@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"status" : "PUBLISHED",
"swapOrderId" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",

```

```

"updateBy": "0833a573fe80d5401f6dc010a775569a",
"updateTime": "2023-11-01T08:18:56.000Z",
"versionId": null,
"versionName": null,
"versionTag": null
},
"childs": [ {
"id": "1169309823469330432",
"name": "原始记录",
"qname": "city_traffic.stroke_reminder.origin_stroke@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"guid": null,
"ordinal": 1,
"row": null,
"catalog": {
"versionName": null,
"versionTag": null,
"versionId": null,
"createBy": "0833a573fe80d5401f6dc010a775569a",
"updateBy": "0833a573fe80d5401f6dc010a775569a",
"createTime": "2023-11-01T08:19:47.000Z",
"updateTime": "2023-11-01T08:23:18.000Z",
"l3Id": null,
"l2Id": null,
"alias": "",
"encoding": null,
"prefix": null,
"codeNum": null,
"id": "1169309823469330432",
"nameCh": "原始记录",
"nameEn": "origin_stroke",
"qualifiedName":
"city_traffic.stroke_reminder.origin_stroke@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description": "无",
"owner": "0833a573fe80d5401f6dc010a775569a",
"guid": "3896ba50-2fd5-4cbc-b428-e58bc4a100f8",
"code": null,
"tenantId": "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields": [ ],
"dataOwner": "test_uesr",
"dataOwnerList": ["test_uesr"],
"dataDepartment": null,
"path": "城市交通/行程记录/原始记录",
"parentId": "1169309374670356480",
"level": 3,
"ordinal": 1,
"status": "PUBLISHED",
"approvalInfo": null,
"newBiz": null,
"fromPublic": false,
"swapOrderId": null,
"levelStr": "3",
"lastL2Id": null
},
"attributes": {
"alias": "",
"approvalInfo": null,
"code": null,
"codeNum": null,
"createBy": "0833a573fe80d5401f6dc010a775569a",
"createTime": "2023-11-01T08:19:47.000Z",
"dataDepartment": null,
"dataOwner": "test_uesr",
"dataOwnerList": [ "test_uesr" ],
"description": "无",
"encoding": null,
"fromPublic": false,

```

```

"guid" : "3896ba50-2fd5-4cbc-b428-e58bc4a100f8",
"id" : "1169309823469330432",
"l1" : false,
"l2" : true,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 3,
"levelStr" : "3",
"name" : "原始记录",
"nameCh" : "原始记录",
"nameEn" : "origin_stroke",
"nameEng" : "origin_stroke",
"newBiz" : null,
"ordinal" : 1,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : "1169309374670356480",
"path" : "城市交通/行程记录/原始记录",
"prefix" : null,
"qualifiedName" :
"city_traffic.stroke_reminder.origin_stroke@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"status" : "PUBLISHED",
"swapOrderId" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"updateTime" : "2023-11-01T08:23:18.000Z",
"versionId" : null,
"versionName" : null,
"versionTag" : null
},
"childs" : [ ],
"count" : 0,
"status" : "PUBLISHED"
}, {
"id" : "1169309948774162432",
"name" : "标准记录",
"qname" : "city_traffic.stroke_reminder.stand_stroke@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"guid" : null,
"ordinal" : 2,
"row" : null,
"catalog" : {
"versionName" : null,
"versionTag" : null,
"versionId" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-01T08:20:17.000Z",
"updateTime" : "2023-11-01T08:23:19.000Z",
"l3Id" : null,
"l2Id" : null,
"alias" : "",
"encoding" : null,
"prefix" : null,
"codeNum" : null,
"id" : "1169309948774162432",
"nameCh" : "标准记录",
"nameEn" : "stand_stroke",
"qualifiedName" :
"city_traffic.stroke_reminder.stand_stroke@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "无",
"owner" : "0833a573fe80d5401f6dc010a775569a",
"guid" : "e674325a-d590-44f5-9723-66b19ba751fd",
"code" : null,

```

```

        "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "selfDefinedFields" : [ ],
        "dataOwner" : "test_uesr",
        "dataOwnerList" : "[\"test_uesr\"]",
        "dataDepartment" : null,
        "path" : "城市交通/行程记录/标准记录",
        "parentId" : "1169309374670356480",
        "level" : 3,
        "ordinal" : 2,
        "status" : "PUBLISHED",
        "approvalInfo" : null,
        "newBiz" : null,
        "fromPublic" : false,
        "swapOrderId" : null,
        "levelStr" : "3",
        "lastL2Id" : null
    },
    "attributes" : {
        "alias" : "",
        "approvalInfo" : null,
        "code" : null,
        "codeNum" : null,
        "createBy" : "0833a573fe80d5401f6dc010a775569a",
        "createTime" : "2023-11-01T08:20:17.000Z",
        "dataDepartment" : null,
        "dataOwner" : "test_uesr",
        "dataOwnerList" : [ "test_uesr" ],
        "description" : "无",
        "encoding" : null,
        "fromPublic" : false,
        "guid" : "e674325a-d590-44f5-9723-66b19ba751fd",
        "id" : "1169309948774162432",
        "l1" : false,
        "l2" : true,
        "l2Id" : null,
        "l3" : false,
        "l3Id" : null,
        "lastL2Id" : null,
        "level" : 3,
        "levelStr" : "3",
        "name" : "标准记录",
        "nameCh" : "标准记录",
        "nameEn" : "stand_stroke",
        "nameEng" : "stand_stroke",
        "newBiz" : null,
        "ordinal" : 2,
        "owner" : "0833a573fe80d5401f6dc010a775569a",
        "parentId" : "1169309374670356480",
        "path" : "城市交通/行程记录/标准记录",
        "prefix" : null,
        "qualifiedName" :
"city_traffic.stroke_reminder.stand_stroke@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "selfDefinedFields" : [ ],
        "status" : "PUBLISHED",
        "swapOrderId" : null,
        "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "updateBy" : "0833a573fe80d5401f6dc010a775569a",
        "updateTime" : "2023-11-01T08:23:19.000Z",
        "versionId" : null,
        "versionName" : null,
        "versionTag" : null
    },
    "childs" : [ ],
    "count" : 0,
    "status" : "PUBLISHED"
}, {

```

```

    "id" : "1169310587398889472",
    "name" : "行程事实",
    "qname" : "city_traffic.stroke_reminder.stroke_fact@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "guid" : null,
    "ordinal" : 3,
    "row" : null,
    "catalog" : {
      "versionName" : null,
      "versionTag" : null,
      "versionId" : null,
      "createBy" : "0833a573fe80d5401f6dc010a775569a",
      "updateBy" : "0833a573fe80d5401f6dc010a775569a",
      "createTime" : "2023-11-01T08:22:49.000Z",
      "updateTime" : "2023-11-01T08:23:20.000Z",
      "l3Id" : null,
      "l2Id" : null,
      "alias" : "",
      "encoding" : null,
      "prefix" : null,
      "codeNum" : null,
      "id" : "1169310587398889472",
      "nameCh" : "行程事实",
      "nameEn" : "stroke_fact",
      "qualifiedName" :
"city_traffic.stroke_reminder.stroke_fact@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "description" : "无",
      "owner" : "0833a573fe80d5401f6dc010a775569a",
      "guid" : "4a396cfa-7b92-417a-af51-a8c33aa230a4",
      "code" : null,
      "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "selfDefinedFields" : [ ],
      "dataOwner" : "test_uesr",
      "dataOwnerList" : ["test_uesr"],
      "dataDepartment" : null,
      "path" : "城市交通/行程记录/行程事实",
      "parentId" : "1169309374670356480",
      "level" : 3,
      "ordinal" : 3,
      "status" : "PUBLISHED",
      "approvalInfo" : null,
      "newBiz" : null,
      "fromPublic" : false,
      "swapOrderId" : null,
      "levelStr" : "3",
      "lastL2Id" : null
    },
    "attributes" : {
      "alias" : "",
      "approvalInfo" : null,
      "code" : null,
      "codeNum" : null,
      "createBy" : "0833a573fe80d5401f6dc010a775569a",
      "createTime" : "2023-11-01T08:22:49.000Z",
      "dataDepartment" : null,
      "dataOwner" : "test_uesr",
      "dataOwnerList" : [ "test_uesr" ],
      "description" : "无",
      "encoding" : null,
      "fromPublic" : false,
      "guid" : "4a396cfa-7b92-417a-af51-a8c33aa230a4",
      "id" : "1169310587398889472",
      "l1" : false,
      "l2" : true,
      "l2Id" : null,
      "l3" : false,
      "l3Id" : null,

```

```

        "lastL2Id" : null,
        "level" : 3,
        "levelStr" : "3",
        "name" : "行程事实",
        "nameCh" : "行程事实",
        "nameEn" : "stroke_fact",
        "nameEng" : "stroke_fact",
        "newBiz" : null,
        "ordinal" : 3,
        "owner" : "0833a573fe80d5401f6dc010a775569a",
        "parentId" : "1169309374670356480",
        "path" : "城市交通/行程记录/行程事实",
        "prefix" : null,
        "qualifiedName" :
"city_traffic.stroke_reminder.stroke_fact@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "selfDefinedFields" : [ ],
        "status" : "PUBLISHED",
        "swapOrderId" : null,
        "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "updateBy" : "0833a573fe80d5401f6dc010a775569a",
        "updateTime" : "2023-11-01T08:23:20.000Z",
        "versionId" : null,
        "versionName" : null,
        "versionTag" : null
    },
    "childs" : [ ],
    "count" : 0,
    "status" : "PUBLISHED"
}, {
    "id" : "1169310688582279168",
    "name" : "记录统计",
    "qname" :
"city_traffic.stroke_reminder.stroke_statistic@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "guid" : null,
    "ordinal" : 4,
    "row" : null,
    "catalog" : {
        "versionName" : null,
        "versionTag" : null,
        "versionId" : null,
        "createBy" : "0833a573fe80d5401f6dc010a775569a",
        "updateBy" : "0833a573fe80d5401f6dc010a775569a",
        "createTime" : "2023-11-01T08:23:13.000Z",
        "updateTime" : "2023-11-01T08:23:20.000Z",
        "l3Id" : null,
        "l2Id" : null,
        "alias" : "",
        "encoding" : null,
        "prefix" : null,
        "codeNum" : null,
        "id" : "1169310688582279168",
        "nameCh" : "记录统计",
        "nameEn" : "stroke_statistic",
        "qualifiedName" :
"city_traffic.stroke_reminder.stroke_statistic@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "description" : "无",
        "owner" : "0833a573fe80d5401f6dc010a775569a",
        "guid" : "ab1bb932-21f7-4d7c-bd83-8950ef4a9adf",
        "code" : null,
        "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "selfDefinedFields" : [ ],
        "dataOwner" : "test_uesr",
        "dataOwnerList" : ["test_uesr"],
        "dataDepartment" : null,

```

```

"path": "城市交通/行程记录/记录统计",
"parentId": "1169309374670356480",
"level": 3,
"ordinal": 4,
"status": "PUBLISHED",
"approvalInfo": null,
"newBiz": null,
"fromPublic": false,
"swapOrderId": null,
"levelStr": "3",
"lastL2Id": null
},
"attributes": {
"alias": "",
"approvalInfo": null,
"code": null,
"codeNum": null,
"createBy": "0833a573fe80d5401f6dc010a775569a",
"createTime": "2023-11-01T08:23:13.000Z",
"dataDepartment": null,
"dataOwner": "test_uesr",
"dataOwnerList": [ "test_uesr" ],
"description": "无",
"encoding": null,
"fromPublic": false,
"guid": "ab1bb932-21f7-4d7c-bd83-8950ef4a9adf",
"id": "1169310688582279168",
"l1": false,
"l2": true,
"l2Id": null,
"l3": false,
"l3Id": null,
"lastL2Id": null,
"level": 3,
"levelStr": "3",
"name": "记录统计",
"nameCh": "记录统计",
"nameEn": "stroke_statistic",
"nameEng": "stroke_statistic",
"newBiz": null,
"ordinal": 4,
"owner": "0833a573fe80d5401f6dc010a775569a",
"parentId": "1169309374670356480",
"path": "城市交通/行程记录/记录统计",
"prefix": null,
"qualifiedName":
"city_traffic.stroke_reminder.stroke_statistic@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields": [ ],
"status": "PUBLISHED",
"swapOrderId": null,
"tenantId": "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy": "0833a573fe80d5401f6dc010a775569a",
"updateTime": "2023-11-01T08:23:20.000Z",
"versionId": null,
"versionName": null,
"versionTag": null
},
"childs": [ ],
"count": 0,
"status": "PUBLISHED"
}],
"count": 4,
"status": "PUBLISHED"
}, {
"id": "1169309441888272384",
"name": "集团",
"qname": "city_traffic.people@Business.0833a5737480d53b2f25c010dc1a7b88-

```



```
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid" : null,
  "ordinal" : 2,
  "row" : null,
  "catalog" : {
    "versionName" : null,
    "versionTag" : null,
    "versionId" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2023-11-01T08:18:16.000Z",
    "updateTime" : "2023-11-01T08:18:57.000Z",
    "l3Id" : null,
    "l2Id" : null,
    "alias" : "",
    "encoding" : null,
    "prefix" : null,
    "codeNum" : null,
    "id" : "1169309441888272384",
    "nameCh" : "集团",
    "nameEn" : "people",
    "qualifiedName" : "city_traffic.people@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : "",
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : "55c88532-26f5-4cc7-bf3c-27d037725e6e",
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "dataOwner" : "",
    "dataOwnerList" : ["test_uesr\"],
    "dataDepartment" : null,
    "path" : "城市交通/集团",
    "parentId" : "1169309252771299328",
    "level" : 2,
    "ordinal" : 2,
    "status" : "PUBLISHED",
    "approvalInfo" : null,
    "newBiz" : null,
    "fromPublic" : false,
    "swapOrderId" : null,
    "levelStr" : "2",
    "lastL2Id" : null
  },
  "attributes" : {
    "alias" : "",
    "approvalInfo" : null,
    "code" : null,
    "codeNum" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2023-11-01T08:18:16.000Z",
    "dataDepartment" : null,
    "dataOwner" : "",
    "dataOwnerList" : [ "test_uesr" ],
    "description" : "",
    "encoding" : null,
    "fromPublic" : false,
    "guid" : "55c88532-26f5-4cc7-bf3c-27d037725e6e",
    "id" : "1169309441888272384",
    "l1" : false,
    "l2" : true,
    "l2Id" : null,
    "l3" : false,
    "l3Id" : null,
    "lastL2Id" : null,
    "level" : 2,
    "levelStr" : "2",
    "name" : "集团",
    "nameCh" : "集团",
```

```

"nameEn": "people",
"nameEng": "people",
"newBiz": null,
"ordinal": 2,
"owner": "0833a573fe80d5401f6dc010a775569a",
"parentId": "1169309252771299328",
"path": "城市交通/集团",
"prefix": null,
"qualifiedName": "city_traffic.people@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields": [ ],
"status": "PUBLISHED",
"swapOrderId": null,
"tenantId": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy": "0833a573fe80d5401f6dc010a775569a",
"updateTime": "2023-11-01T08:18:57.000Z",
"versionId": null,
"versionName": null,
"versionTag": null
},
"childs": [ {
"id": "1169310842425155584",
"name": "供应商",
"qname": "city_traffic.people.vendor@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"guid": null,
"ordinal": 1,
"row": null,
"catalog": {
"versionName": null,
"versionTag": null,
"versionId": null,
"createBy": "0833a573fe80d5401f6dc010a775569a",
"updateBy": "0833a573fe80d5401f6dc010a775569a",
"createTime": "2023-11-01T08:23:50.000Z",
"updateTime": "2023-11-01T08:25:01.000Z",
"l3Id": null,
"l2Id": null,
"alias": "",
"encoding": null,
"prefix": null,
"codeNum": null,
"id": "1169310842425155584",
"nameCh": "供应商",
"nameEn": "vendor",
"qualifiedName": "city_traffic.people.vendor@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description": "无",
"owner": "0833a573fe80d5401f6dc010a775569a",
"guid": "0dfb1ec3-6df6-475c-806f-36e5872346e2",
"code": null,
"tenantId": "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields": [ ],
"dataOwner": "test_uesr",
"dataOwnerList": ["test_uesr"],
"dataDepartment": null,
"path": "城市交通/集团/供应商",
"parentId": "1169309441888272384",
"level": 3,
"ordinal": 1,
"status": "PUBLISHED",
"approvalInfo": null,
"newBiz": null,
"fromPublic": false,
"swapOrderId": null,
"levelStr": "3",
"lastL2Id": null
},
},

```

```

"attributes" : {
  "alias" : "",
  "approvalInfo" : null,
  "code" : null,
  "codeNum" : null,
  "createBy" : "0833a573fe80d5401f6dc010a775569a",
  "createTime" : "2023-11-01T08:23:50.000Z",
  "dataDepartment" : null,
  "dataOwner" : "test_uesr",
  "dataOwnerList" : [ "test_uesr" ],
  "description" : "无",
  "encoding" : null,
  "fromPublic" : false,
  "guid" : "0dfb1ec3-6df6-475c-806f-36e5872346e2",
  "id" : "1169310842425155584",
  "l1" : false,
  "l2" : true,
  "l2Id" : null,
  "l3" : false,
  "l3Id" : null,
  "lastL2Id" : null,
  "level" : 3,
  "levelStr" : "3",
  "name" : "供应商",
  "nameCh" : "供应商",
  "nameEn" : "vendor",
  "nameEng" : "vendor",
  "newBiz" : null,
  "ordinal" : 1,
  "owner" : "0833a573fe80d5401f6dc010a775569a",
  "parentId" : "1169309441888272384",
  "path" : "城市交通/集团/供应商",
  "prefix" : null,
  "qualifiedName" : "city_traffic.people.vendor@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "selfDefinedFields" : [ ],
  "status" : "PUBLISHED",
  "swapOrderId" : null,
  "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "updateBy" : "0833a573fe80d5401f6dc010a775569a",
  "updateTime" : "2023-11-01T08:25:01.000Z",
  "versionId" : null,
  "versionName" : null,
  "versionTag" : null
},
"childs" : [ ],
"count" : 0,
"status" : "PUBLISHED"
}],
"count" : 1,
"status" : "PUBLISHED"
}, {
  "id" : "1169309513090834432",
  "name" : "时空",
  "qname" : "city_traffic.time_location@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid" : null,
  "ordinal" : 3,
  "row" : null,
  "catalog" : {
    "versionName" : null,
    "versionTag" : null,
    "versionId" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2023-11-01T08:18:33.000Z",
    "updateTime" : "2023-11-01T08:18:57.000Z",
    "l3Id" : null,

```

```

    "l2Id" : null,
    "alias" : "",
    "encoding" : null,
    "prefix" : null,
    "codeNum" : null,
    "id" : "1169309513090834432",
    "nameCh" : "时空",
    "nameEn" : "time_location",
    "qualifiedName" : "city_traffic.time_location@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : "",
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : "5025be1c-b217-4d4e-8927-78f2acd1700f",
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "dataOwner" : "",
    "dataOwnerList" : ["\test_uesr\"],
    "dataDepartment" : null,
    "path" : "城市交通/时空",
    "parentId" : "1169309252771299328",
    "level" : 2,
    "ordinal" : 3,
    "status" : "PUBLISHED",
    "approvalInfo" : null,
    "newBiz" : null,
    "fromPublic" : false,
    "swapOrderId" : null,
    "levelStr" : "2",
    "lastL2Id" : null
  },
  "attributes" : {
    "alias" : "",
    "approvalInfo" : null,
    "code" : null,
    "codeNum" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2023-11-01T08:18:33.000Z",
    "dataDepartment" : null,
    "dataOwner" : "",
    "dataOwnerList" : [ "test_uesr" ],
    "description" : "",
    "encoding" : null,
    "fromPublic" : false,
    "guid" : "5025be1c-b217-4d4e-8927-78f2acd1700f",
    "id" : "1169309513090834432",
    "l1" : false,
    "l2" : true,
    "l2Id" : null,
    "l3" : false,
    "l3Id" : null,
    "lastL2Id" : null,
    "level" : 2,
    "levelStr" : "2",
    "name" : "时空",
    "nameCh" : "时空",
    "nameEn" : "time_location",
    "nameEng" : "time_location",
    "newBiz" : null,
    "ordinal" : 3,
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "parentId" : "1169309252771299328",
    "path" : "城市交通/时空",
    "prefix" : null,
    "qualifiedName" : "city_traffic.time_location@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "status" : "PUBLISHED",
    "swapOrderId" : null,

```

```

"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"updateTime" : "2023-11-01T08:18:57.000Z",
"versionId" : null,
"versionName" : null,
"versionTag" : null
},
"childs" : [ {
  "id" : "1169310937660964864",
  "name" : "时间",
  "qname" : "city_traffic.time_location.date@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid" : null,
  "ordinal" : 1,
  "row" : null,
  "catalog" : {
    "versionName" : null,
    "versionTag" : null,
    "versionId" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2023-11-01T08:24:12.000Z",
    "updateTime" : "2023-11-01T08:25:12.000Z",
    "l3Id" : null,
    "l2Id" : null,
    "alias" : "",
    "encoding" : null,
    "prefix" : null,
    "codeNum" : null,
    "id" : "1169310937660964864",
    "nameCh" : "时间",
    "nameEn" : "date",
    "qualifiedName" : "city_traffic.time_location.date@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : "无",
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : "b3a35c3a-87c6-4316-ac84-76fa848f5ce6",
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "dataOwner" : "test_uesr",
    "dataOwnerList" : ["test_uesr"],
    "dataDepartment" : null,
    "path" : "城市交通/时空/时间",
    "parentId" : "1169309513090834432",
    "level" : 3,
    "ordinal" : 1,
    "status" : "PUBLISHED",
    "approvalInfo" : null,
    "newBiz" : null,
    "fromPublic" : false,
    "swapOrderId" : null,
    "levelStr" : "3",
    "lastL2Id" : null
  },
  "attributes" : {
    "alias" : "",
    "approvalInfo" : null,
    "code" : null,
    "codeNum" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2023-11-01T08:24:12.000Z",
    "dataDepartment" : null,
    "dataOwner" : "test_uesr",
    "dataOwnerList" : [ "test_uesr" ],
    "description" : "无",
    "encoding" : null,
    "fromPublic" : false,

```

```

"guid" : "b3a35c3a-87c6-4316-ac84-76fa848f5ce6",
"id" : "1169310937660964864",
"l1" : false,
"l2" : true,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 3,
"levelStr" : "3",
"name" : "时间",
"nameCh" : "时间",
"nameEn" : "date",
"nameEng" : "date",
"newBiz" : null,
"ordinal" : 1,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : "1169309513090834432",
"path" : "城市交通/时空/时间",
"prefix" : null,
"qualifiedName" : "city_traffic.time_location.date@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"status" : "PUBLISHED",
"swapOrderId" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"updateTime" : "2023-11-01T08:25:12.000Z",
"versionId" : null,
"versionName" : null,
"versionTag" : null
},
"childs" : [ ],
"count" : 0,
"status" : "PUBLISHED"
}],
"count" : 1,
"status" : "PUBLISHED"
}, {
"id" : "1169309574931652608",
"name" : "公共维度",
"qname" : "city_traffic.public_dimension@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"guid" : null,
"ordinal" : 4,
"row" : null,
"catalog" : {
"versionName" : null,
"versionTag" : null,
"versionId" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-01T08:18:47.000Z",
"updateTime" : "2023-11-01T08:18:58.000Z",
"l3Id" : null,
"l2Id" : null,
"alias" : "",
"encoding" : null,
"prefix" : null,
"codeNum" : null,
"id" : "1169309574931652608",
"nameCh" : "公共维度",
"nameEn" : "public_dimension",
"qualifiedName" : "city_traffic.public_dimension@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "",
"owner" : "0833a573fe80d5401f6dc010a775569a",
"guid" : "7f1038f1-66c7-4002-b53e-f9dcf429fd9",

```

```

"code" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"dataOwner" : "",
"dataOwnerList" : ["\test_uesr\""],
"dataDepartment" : null,
"path" : "城市交通/公共维度",
"parentId" : "1169309252771299328",
"level" : 2,
"ordinal" : 4,
"status" : "PUBLISHED",
"approvalInfo" : null,
"newBiz" : null,
"fromPublic" : false,
"swapOrderId" : null,
"levelStr" : "2",
"lastL2Id" : null
},
"attributes" : {
"alias" : "",
"approvalInfo" : null,
"code" : null,
"codeNum" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-01T08:18:47.000Z",
"dataDepartment" : null,
"dataOwner" : "",
"dataOwnerList" : [ "test_uesr" ],
"description" : "",
"encoding" : null,
"fromPublic" : false,
"guid" : "7f1038f1-66c7-4002-b53e-f9dcf429fdf9",
"id" : "1169309574931652608",
"l1" : false,
"l2" : true,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 2,
"levelStr" : "2",
"name" : "公共维度",
"nameCh" : "公共维度",
"nameEn" : "public_dimension",
"nameEng" : "public_dimension",
"newBiz" : null,
"ordinal" : 4,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : "1169309252771299328",
"path" : "城市交通/公共维度",
"prefix" : null,
"qualifiedName" : "city_traffic.public_dimension@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"status" : "PUBLISHED",
"swapOrderId" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"updateTime" : "2023-11-01T08:18:58.000Z",
"versionId" : null,
"versionName" : null,
"versionTag" : null
},
"childs" : [ {
"id" : "1169311054568800256",
"name" : "公共维度",
"qname" :
"city_traffic.public_dimension.public_dimension@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",

```

```

"guid" : null,
"ordinal" : 1,
"row" : null,
"catalog" : {
  "versionName" : null,
  "versionTag" : null,
  "versionId" : null,
  "createBy" : "0833a573fe80d5401f6dc010a775569a",
  "updateBy" : "0833a573fe80d5401f6dc010a775569a",
  "createTime" : "2023-11-01T08:24:40.000Z",
  "updateTime" : "2023-11-01T08:25:22.000Z",
  "l3Id" : null,
  "l2Id" : null,
  "alias" : "",
  "encoding" : null,
  "prefix" : null,
  "codeNum" : null,
  "id" : "1169311054568800256",
  "nameCh" : "公共维度",
  "nameEn" : "public_dimension",
  "qualifiedName" :
"city_traffic.public_dimension.public_dimension@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "description" : "无",
  "owner" : "0833a573fe80d5401f6dc010a775569a",
  "guid" : "eb7d8e4f-1a0d-4a59-9842-429af5efef48",
  "code" : null,
  "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "selfDefinedFields" : [ ],
  "dataOwner" : "test_uesr",
  "dataOwnerList" : ["test_uesr"],
  "dataDepartment" : null,
  "path" : "城市交通/公共维度/公共维度",
  "parentId" : "1169309574931652608",
  "level" : 3,
  "ordinal" : 1,
  "status" : "PUBLISHED",
  "approvalInfo" : null,
  "newBiz" : null,
  "fromPublic" : false,
  "swapOrderId" : null,
  "levelStr" : "3",
  "lastL2Id" : null
},
"attributes" : {
  "alias" : "",
  "approvalInfo" : null,
  "code" : null,
  "codeNum" : null,
  "createBy" : "0833a573fe80d5401f6dc010a775569a",
  "createTime" : "2023-11-01T08:24:40.000Z",
  "dataDepartment" : null,
  "dataOwner" : "test_uesr",
  "dataOwnerList" : [ "test_uesr" ],
  "description" : "无",
  "encoding" : null,
  "fromPublic" : false,
  "guid" : "eb7d8e4f-1a0d-4a59-9842-429af5efef48",
  "id" : "1169311054568800256",
  "l1" : false,
  "l2" : true,
  "l2Id" : null,
  "l3" : false,
  "l3Id" : null,
  "lastL2Id" : null,
  "level" : 3,
  "levelStr" : "3",
  "name" : "公共维度",

```



```

"nameCh": "公共维度",
"nameEn": "public_dimension",
"nameEng": "public_dimension",
"newBiz": null,
"ordinal": 1,
"owner": "0833a573fe80d5401f6dc010a775569a",
"parentId": "1169309574931652608",
"path": "城市交通/公共维度/公共维度",
"prefix": null,
"qualifiedName":
"city_traffic.public_dimension.public_dimension@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields": [ ],
"status": "PUBLISHED",
"swapOrderId": null,
"tenantId": "0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy": "0833a573fe80d5401f6dc010a775569a",
"updateTime": "2023-11-01T08:25:22.000Z",
"versionId": null,
"versionName": null,
"versionTag": null
},
"childs": [ ],
"count": 0,
"status": "PUBLISHED"
}],
"count": 1,
"status": "PUBLISHED"
}],
"count": 4,
"status": "PUBLISHED"
}, {
"id": "1170064658061799424",
"name": "测试主题",
"qname": "test_subject@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"guid": null,
"ordinal": 3,
"row": null,
"catalog": {
"versionName": null,
"versionTag": null,
"versionId": null,
"createBy": "0833a573fe80d5401f6dc010a775569a",
"updateBy": "0833a573fe80d5401f6dc010a775569a",
"createTime": "2023-11-03T10:19:13.000Z",
"updateTime": "2023-11-03T10:28:07.000Z",
"l3Id": null,
"l2Id": null,
"alias": "",
"encoding": null,
"prefix": null,
"codeNum": null,
"id": "1170064658061799424",
"nameCh": "测试主题",
"nameEn": "test_subject",
"qualifiedName": "test_subject@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description": "",
"owner": "0833a573fe80d5401f6dc010a775569a",
"guid": "8c930e34-0003-4999-ab79-66374435d980",
"code": null,
"tenantId": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields": [ ],
"dataOwner": "",
"dataOwnerList": "[\"test_ues\"]",
"dataDepartment": null,
"path": "测试主题",

```

```

"parentId" : null,
"level" : 1,
"ordinal" : 3,
"status" : "PUBLISHED",
"approvalInfo" : null,
"newBiz" : null,
"fromPublic" : false,
"swapOrderId" : null,
"levelStr" : "1",
"lastL2Id" : null
},
"attributes" : {
"alias" : "",
"approvalInfo" : null,
"code" : null,
"codeNum" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-03T10:19:13.000Z",
"dataDepartment" : null,
"dataOwner" : "",
"dataOwnerList" : [ "test_uesr" ],
"description" : "",
"encoding" : null,
"fromPublic" : false,
"guid" : "8c930e34-0003-4999-ab79-66374435d980",
"id" : "1170064658061799424",
"l1" : true,
"l2" : false,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 1,
"levelStr" : "1",
"name" : "测试主题",
"nameCh" : "测试主题",
"nameEn" : "test_subject",
"nameEng" : "test_subject",
"newBiz" : null,
"ordinal" : 3,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : null,
"path" : "测试主题",
"prefix" : null,
"qualifiedName" : "test_subject@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"status" : "PUBLISHED",
"swapOrderId" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"updateTime" : "2023-11-03T10:28:07.000Z",
"versionId" : null,
"versionName" : null,
"versionTag" : null
},
"childs" : [ ],
"count" : 0,
"status" : "PUBLISHED"
}, {
"id" : "1170066794946420736",
"name" : "test_001",
"qname" : "test_czh.BO000000@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"guid" : null,
"ordinal" : 4,
"row" : null,
"catalog" : {
"versionName" : null,

```

```

"versionTag" : null,
"versionId" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-03T10:27:43.000Z",
"updateTime" : "2023-11-03T10:29:02.000Z",
"l3Id" : null,
"l2Id" : null,
"alias" : "",
"encoding" : null,
"prefix" : null,
"codeNum" : null,
"id" : "1170066794946420736",
"nameCh" : "test_001",
"nameEn" : "BO000000",
"qualifiedName" : "test_czh.BO000000@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "无",
"owner" : "0833a573fe80d5401f6dc010a775569a",
"guid" : "99e0ec15-8fc0-4617-b867-e17a0b4a2509",
"code" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"dataOwner" : "1",
"dataOwnerList" : ["test_uesr"],
"dataDepartment" : null,
"path" : "test_001",
"parentId" : null,
"level" : 1,
"ordinal" : 4,
"status" : "PUBLISHED",
"approvalInfo" : null,
"newBiz" : null,
"fromPublic" : false,
"swapOrderId" : null,
"levelStr" : "1",
"lastL2Id" : null
},
"attributes" : {
"alias" : "",
"approvalInfo" : null,
"code" : null,
"codeNum" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-03T10:27:43.000Z",
"dataDepartment" : null,
"dataOwner" : "1",
"dataOwnerList" : [ "test_uesr" ],
"description" : "无",
"encoding" : null,
"fromPublic" : false,
"guid" : "99e0ec15-8fc0-4617-b867-e17a0b4a2509",
"id" : "1170066794946420736",
"l1" : true,
"l2" : false,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 1,
"levelStr" : "1",
"name" : "test_001",
"nameCh" : "test_001",
"nameEn" : "BO000000",
"nameEng" : "BO000000",
"newBiz" : null,
"ordinal" : 4,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : null,

```

```

    "path": "test_001",
    "prefix": null,
    "qualifiedName": "test_czh.BO000000@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields": [ ],
    "status": "PUBLISHED",
    "swapOrderId": null,
    "tenantId": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "updateBy": "0833a573fe80d5401f6dc010a775569a",
    "updateTime": "2023-11-03T10:29:02.000Z",
    "versionId": null,
    "versionName": null,
    "versionTag": null
  },
  "childs": [ ],
  "count": 0,
  "status": "PUBLISHED"
}, {
  "id": "1178033294286176256",
  "name": "我的测试",
  "qname": "MY_000001@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid": null,
  "ordinal": 5,
  "row": null,
  "catalog": {
    "versionName": null,
    "versionTag": null,
    "versionId": null,
    "createBy": "0833a573fe80d5401f6dc010a775569a",
    "updateBy": "0833a573fe80d5401f6dc010a775569a",
    "createTime": "2023-11-25T10:03:44.000Z",
    "updateTime": "2023-11-30T08:07:34.000Z",
    "l3Id": null,
    "l2Id": null,
    "alias": "",
    "encoding": null,
    "prefix": null,
    "codeNum": null,
    "id": "1178033294286176256",
    "nameCh": "我的测试",
    "nameEn": "MY_000001",
    "qualifiedName": "MY_000001@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description": "",
    "owner": "0833a573fe80d5401f6dc010a775569a",
    "guid": "263b2018-c360-4718-9150-3f1fc67bb97e",
    "code": null,
    "tenantId": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields": [ ],
    "dataOwner": "",
    "dataOwnerList": "[\"test_ues\"]",
    "dataDepartment": null,
    "path": "我的测试",
    "parentId": null,
    "level": 1,
    "ordinal": 5,
    "status": "PUBLISHED",
    "approvalInfo": null,
    "newBiz": null,
    "fromPublic": false,
    "swapOrderId": null,
    "levelStr": "1",
    "lastL2Id": null
  },
  "attributes": {
    "alias": "",
    "approvalInfo": null,
    "code": null,

```

```

"codeNum" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-25T10:03:44.000Z",
"dataDepartment" : null,
"dataOwner" : "",
"dataOwnerList" : [ "test_uesr" ],
"description" : "",
"encoding" : null,
"fromPublic" : false,
"guid" : "263b2018-c360-4718-9150-3f1fc67bb97e",
"id" : "1178033294286176256",
"l1" : true,
"l2" : false,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 1,
"levelStr" : "1",
"name" : "我的测试",
"nameCh" : "我的测试",
"nameEn" : "MY_000001",
"nameEng" : "MY_000001",
"newBiz" : null,
"ordinal" : 5,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : null,
"path" : "我的测试",
"prefix" : null,
"qualifiedName" : "MY_000001@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"status" : "PUBLISHED",
"swapOrderId" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"updateTime" : "2023-11-30T08:07:34.000Z",
"versionId" : null,
"versionName" : null,
"versionTag" : null
},
"childs" : [ ],
"count" : 0,
"status" : "PUBLISHED"
}, {
"id" : "1170064724101115904",
"name" : "test_czh2",
"qname" : "test_czh@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"guid" : null,
"ordinal" : 6,
"row" : null,
"catalog" : {
"versionName" : null,
"versionTag" : null,
"versionId" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2023-11-03T10:19:29.000Z",
"updateTime" : "2024-02-22T09:21:33.000Z",
"l3Id" : null,
"l2Id" : null,
"alias" : "",
"encoding" : null,
"prefix" : null,
"codeNum" : null,
"id" : "1170064724101115904",
"nameCh" : "test_czh2",
"nameEn" : "test_czh",

```

```

    "qualifiedName": "test_czh@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description": "",
    "owner": "0833a573fe80d5401f6dc010a775569a",
    "guid": "0941628c-af28-4f82-9bdd-2f101ca70f3a",
    "code": null,
    "tenantId": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields": [ ],
    "dataOwner": "",
    "dataOwnerList": ["test_uesr"],
    "dataDepartment": null,
    "path": "test_czh2",
    "parentId": null,
    "level": 1,
    "ordinal": 6,
    "status": "PUBLISHED",
    "approvalInfo": null,
    "newBiz": null,
    "fromPublic": false,
    "swapOrderId": null,
    "levelStr": "1",
    "lastL2Id": null
  },
  "attributes": {
    "alias": "",
    "approvalInfo": null,
    "code": null,
    "codeNum": null,
    "createBy": "0833a573fe80d5401f6dc010a775569a",
    "createTime": "2023-11-03T10:19:29.000Z",
    "dataDepartment": null,
    "dataOwner": "",
    "dataOwnerList": [ "test_uesr" ],
    "description": "",
    "encoding": null,
    "fromPublic": false,
    "guid": "0941628c-af28-4f82-9bdd-2f101ca70f3a",
    "id": "1170064724101115904",
    "l1": true,
    "l2": false,
    "l2Id": null,
    "l3": false,
    "l3Id": null,
    "lastL2Id": null,
    "level": 1,
    "levelStr": "1",
    "name": "test_czh2",
    "nameCh": "test_czh2",
    "nameEn": "test_czh",
    "nameEng": "test_czh",
    "newBiz": null,
    "ordinal": 6,
    "owner": "0833a573fe80d5401f6dc010a775569a",
    "parentId": null,
    "path": "test_czh2",
    "prefix": null,
    "qualifiedName": "test_czh@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields": [ ],
    "status": "PUBLISHED",
    "swapOrderId": null,
    "tenantId": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "updateBy": "0833a573fe80d5401f6dc010a775569a",
    "updateTime": "2024-02-22T09:21:33.000Z",
    "versionId": null,
    "versionName": null,
    "versionTag": null
  },
  "childs": [ ],

```

```

"count" : 0,
"status" : "PUBLISHED"
}, {
  "id" : "1193143623198326784",
  "name" : "test_import_l2",
  "qname" : "test_import_l2@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid" : null,
  "ordinal" : 7,
  "row" : null,
  "catalog" : {
    "versionName" : null,
    "versionTag" : null,
    "versionId" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2024-01-06T02:46:47.000Z",
    "updateTime" : "2024-01-26T03:17:43.000Z",
    "l3Id" : null,
    "l2Id" : null,
    "alias" : null,
    "encoding" : null,
    "prefix" : null,
    "codeNum" : null,
    "id" : "1193143623198326784",
    "nameCh" : "test_import_l2",
    "nameEn" : "test_import_l2",
    "qualifiedName" : "test_import_l2@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : "ssss",
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : "800a4e28-db08-434d-8f44-877902f2eb55",
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : null,
    "dataOwner" : "ssss",
    "dataOwnerList" : ["test_uesr"],
    "dataDepartment" : null,
    "path" : "test_import_l2",
    "parentId" : null,
    "level" : 1,
    "ordinal" : 7,
    "status" : "PUBLISHED",
    "approvalInfo" : null,
    "newBiz" : null,
    "fromPublic" : false,
    "swapOrderId" : null,
    "levelStr" : "1",
    "lastL2Id" : null
  },
  "attributes" : {
    "alias" : null,
    "approvalInfo" : null,
    "code" : null,
    "codeNum" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2024-01-06T02:46:47.000Z",
    "dataDepartment" : null,
    "dataOwner" : "ssss",
    "dataOwnerList" : [ "test_uesr" ],
    "description" : "ssss",
    "encoding" : null,
    "fromPublic" : false,
    "guid" : "800a4e28-db08-434d-8f44-877902f2eb55",
    "id" : "1193143623198326784",
    "l1" : true,
    "l2" : false,
    "l2Id" : null,
    "l3" : false,

```

```

    "l3Id" : null,
    "lastL2Id" : null,
    "level" : 1,
    "levelStr" : "1",
    "name" : "test_import_l2",
    "nameCh" : "test_import_l2",
    "nameEn" : "test_import_l2",
    "nameEng" : "test_import_l2",
    "newBiz" : null,
    "ordinal" : 7,
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "parentId" : null,
    "path" : "test_import_l2",
    "prefix" : null,
    "qualifiedName" : "test_import_l2@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : null,
    "status" : "PUBLISHED",
    "swapOrderId" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateTime" : "2024-01-26T03:17:43.000Z",
    "versionId" : null,
    "versionName" : null,
    "versionTag" : null
  },
  "childs" : [],
  "count" : 0,
  "status" : "PUBLISHED"
}, {
  "id" : "1193879014310322176",
  "name" : "test_not_publish",
  "qname" : "test_not_publish@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid" : null,
  "ordinal" : 8,
  "row" : null,
  "catalog" : {
    "versionName" : null,
    "versionTag" : null,
    "versionId" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2024-01-08T03:28:58.000Z",
    "updateTime" : "2024-01-08T08:50:51.000Z",
    "l3Id" : null,
    "l2Id" : null,
    "alias" : "",
    "encoding" : null,
    "prefix" : null,
    "codeNum" : null,
    "id" : "1193879014310322176",
    "nameCh" : "test_not_publish",
    "nameEn" : "test_not_publish",
    "qualifiedName" : "test_not_publish@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : "",
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : "df26f160-0692-42dd-95b4-e66fcadca759",
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [],
    "dataOwner" : "",
    "dataOwnerList" : "["test_ues\\"]",
    "dataDepartment" : null,
    "path" : "test_not_publish",
    "parentId" : null,
    "level" : 1,
    "ordinal" : 8,

```



```

        "status": "PUBLISHED",
        "approvalInfo": null,
        "newBiz": null,
        "fromPublic": false,
        "swapOrderId": null,
        "levelStr": "1",
        "lastL2Id": null
    },
    "attributes": {
        "alias": "",
        "approvalInfo": null,
        "code": null,
        "codeNum": null,
        "createBy": "0833a573fe80d5401f6dc010a775569a",
        "createTime": "2024-01-08T03:28:58.000Z",
        "dataDepartment": null,
        "dataOwner": "",
        "dataOwnerList": [ "test_uesr" ],
        "description": "",
        "encoding": null,
        "fromPublic": false,
        "guid": "df26f160-0692-42dd-95b4-e66fcadca759",
        "id": "1193879014310322176",
        "l1": true,
        "l2": false,
        "l2Id": null,
        "l3": false,
        "l3Id": null,
        "lastL2Id": null,
        "level": 1,
        "levelStr": "1",
        "name": "test_not_publish",
        "nameCh": "test_not_publish",
        "nameEn": "test_not_publish",
        "nameEng": "test_not_publish",
        "newBiz": null,
        "ordinal": 8,
        "owner": "0833a573fe80d5401f6dc010a775569a",
        "parentId": null,
        "path": "test_not_publish",
        "prefix": null,
        "qualifiedName": "test_not_publish@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "selfDefinedFields": [ ],
        "status": "PUBLISHED",
        "swapOrderId": null,
        "tenantId": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "updateBy": "0833a573fe80d5401f6dc010a775569a",
        "updateTime": "2024-01-08T08:50:51.000Z",
        "versionId": null,
        "versionName": null,
        "versionTag": null
    },
    "childs": [ ],
    "count": 0,
    "status": "PUBLISHED"
}, {
    "id": "1194966710038474752",
    "name": "test_czh_catalog",
    "qname": "test_czh_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "guid": null,
    "ordinal": 9,
    "row": null,
    "catalog": {
        "versionName": null,
        "versionTag": null,
        "versionId": null,
        "createBy": "0833a573fe80d5401f6dc010a775569a",

```

```

"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2024-01-11T03:31:05.000Z",
"updateTime" : "2024-01-11T03:31:41.000Z",
"l3Id" : null,
"l2Id" : null,
"alias" : "",
"encoding" : null,
"prefix" : null,
"codeNum" : null,
"id" : "1194966710038474752",
"nameCh" : "test_czh_catalog",
"nameEn" : "test_czh_catalog",
"qualifiedName" : "test_czh_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"description" : "",
"owner" : "0833a573fe80d5401f6dc010a775569a",
"guid" : "7d527cb2-9c9b-4a37-8785-04e836095cb1",
"code" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"dataOwner" : "",
"dataOwnerList" : ["test_uesr"],
"dataDepartment" : null,
"path" : "test_czh_catalog",
"parentId" : null,
"level" : 1,
"ordinal" : 9,
"status" : "PUBLISHED",
"approvalInfo" : null,
"newBiz" : null,
"fromPublic" : false,
"swapOrderId" : null,
"levelStr" : "1",
"lastL2Id" : null
},
"attributes" : {
"alias" : "",
"approvalInfo" : null,
"code" : null,
"codeNum" : null,
"createBy" : "0833a573fe80d5401f6dc010a775569a",
"createTime" : "2024-01-11T03:31:05.000Z",
"dataDepartment" : null,
"dataOwner" : "",
"dataOwnerList" : [ "test_uesr" ],
"description" : "",
"encoding" : null,
"fromPublic" : false,
"guid" : "7d527cb2-9c9b-4a37-8785-04e836095cb1",
"id" : "1194966710038474752",
"l1" : true,
"l2" : false,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 1,
"levelStr" : "1",
"name" : "test_czh_catalog",
"nameCh" : "test_czh_catalog",
"nameEn" : "test_czh_catalog",
"nameEng" : "test_czh_catalog",
"newBiz" : null,
"ordinal" : 9,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : null,
"path" : "test_czh_catalog",
"prefix" : null,
"qualifiedName" : "test_czh_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-

```

```

workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "selfDefinedFields" : [ ],
  "status" : "PUBLISHED",
  "swapOrderId" : null,
  "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "updateBy" : "0833a573fe80d5401f6dc010a775569a",
  "updateTime" : "2024-01-11T03:31:41.000Z",
  "versionId" : null,
  "versionName" : null,
  "versionTag" : null
},
"childs" : [ ],
"count" : 0,
"status" : "PUBLISHED"
}, {
  "id" : "1231174343611035648",
  "name" : "test_update_catalog",
  "qname" : "test_create_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "guid" : null,
  "ordinal" : 10,
  "row" : null,
  "catalog" : {
    "versionName" : null,
    "versionTag" : null,
    "versionId" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "updateBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2024-04-20T01:27:18.000Z",
    "updateTime" : "2024-04-20T02:09:55.000Z",
    "l3Id" : null,
    "l2Id" : null,
    "alias" : "",
    "encoding" : null,
    "prefix" : null,
    "codeNum" : null,
    "id" : "1231174343611035648",
    "nameCh" : "test_update_catalog",
    "nameEn" : "test_create_catalog",
    "qualifiedName" : "test_create_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "description" : "this is a update demo.",
    "owner" : "0833a573fe80d5401f6dc010a775569a",
    "guid" : null,
    "code" : null,
    "tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "selfDefinedFields" : [ ],
    "dataOwner" : "dataOwner",
    "dataOwnerList" : "[\"test_ues\"]",
    "dataDepartment" : null,
    "path" : "test_update_catalog",
    "parentId" : null,
    "level" : 1,
    "ordinal" : 10,
    "status" : "DRAFT",
    "approvalInfo" : null,
    "newBiz" : null,
    "fromPublic" : false,
    "swapOrderId" : null,
    "levelStr" : "1",
    "lastL2Id" : null
  },
  "attributes" : {
    "alias" : "",
    "approvalInfo" : null,
    "code" : null,
    "codeNum" : null,
    "createBy" : "0833a573fe80d5401f6dc010a775569a",
    "createTime" : "2024-04-20T01:27:18.000Z",

```

```
"dataDepartment" : null,
"dataOwner" : "dataOwner",
"dataOwnerList" : [ "test_uesr" ],
"description" : "this is a update demo.",
"encoding" : null,
"fromPublic" : false,
"guid" : null,
"id" : "1231174343611035648",
"l1" : true,
"l2" : false,
"l2Id" : null,
"l3" : false,
"l3Id" : null,
"lastL2Id" : null,
"level" : 1,
"levelStr" : "1",
"name" : "test_update_catalog",
"nameCh" : "test_update_catalog",
"nameEn" : "test_create_catalog",
"nameEng" : "test_create_catalog",
"newBiz" : null,
"ordinal" : 10,
"owner" : "0833a573fe80d5401f6dc010a775569a",
"parentId" : null,
"path" : "test_update_catalog",
"prefix" : null,
"qualifiedName" : "test_create_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"selfDefinedFields" : [ ],
"status" : "DRAFT",
"swapOrderId" : null,
"tenantId" : "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
"updateBy" : "0833a573fe80d5401f6dc010a775569a",
"updateTime" : "2024-04-20T02:09:55.000Z",
"versionId" : null,
"versionName" : null,
"versionTag" : null
},
"childs" : [ ],
"count" : 0,
"status" : "DRAFT"
}
}
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
```

```
"error_msg": "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListBusinessSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListBusinessRequest request = new ListBusinessRequest();  
        try {  
            ListBusinessResponse response = client.listBusiness(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBusinessRequest()
        response = client.list_business(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBusinessRequest{}
    response, err := client.ListBusiness(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
```

```
    fmt.Println(err)
  }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是所有l1、l2、l3实体。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.9.6 查找主题列表(新)

### 功能介绍

查找主题(新)。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/design/subjects

表 8-504 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-505 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
name_ch	否	String	按名称精确查询。
name_en	否	String	按编码精确查询。

参数	是否必选	参数类型	描述
create_by	否	String	按创建者查询。
owner	否	String	按负责人查询。
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。
parent_id	否	String	父目录ID，根节点没有此ID，空值为所有，-1为根节点下节点。ID字符串。



## 请求参数

表 8-506 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-507 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-508 data

参数	参数类型	描述
value	<a href="#">value</a> object	value，统一的返回结果的外层数据结构。

表 8-509 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <a href="#">CatalogVO</a> objects	CatalogVO信息。

表 8-510 CatalogVO

参数	参数类型	描述
name_ch	String	中文名称。
name_en	String	英文名称。
description	String	描述信息。
qualified_name	String	扩展名。
guid	String	guid, 自动生成。
code	String	编码。
alias	String	别名。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
new_biz	<a href="#">BizVersionManageVO</a> object	业务版本管理, 只读。
data_owner	String	数据所有者。
data_owner_list	String	数据所有者集合。
data_department	String	数据域。
path	String	路径信息。
level	Integer	层级信息。
ordinal	Integer	序号。
owner	String	责任人。

参数	参数类型	描述
parent_id	String	父目录ID，没有则为根目录，ID字符串。
swap_order_id	String	同层排序，目标节点的ID，ID字符串。
id	String	主题ID，ID字符串。
qualified_id	String	认证ID，自动生成。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
children_num	Integer	拥有子流程的数量，不包括子流程的子流程，前端不传。
children	Array of <b>CatalogVO</b> objects	下层子目录，只读。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	属性自定义项。

表 8-511 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-512 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-513 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-514 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-515 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-516 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查找全部主题。

```
GET https://{endpoint}/v2/{project_id}/design/subjects?parent_id=-1
```

## 响应示例

**状态码：200**

Success，返回data数据是CatalogVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "name_ch": "test_import",
        "name_en": "test_import",
        "description": null,
        "qualified_name": "test_import@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
        "guid": "36999ee2-80df-4cad-b5d5-10c50fbeb401",
        "code": null,
        "status": "PUBLISHED",
        "new_biz": null,
        "alias": null,
        "data_owner": "",
        "data_owner_list": ["test_uesr"],
        "data_department": null,
        "path": "test_import",
        "level": 1,
        "ordinal": 1,
        "owner": "test_uesr",
        "parent_id": null,
        "swap_order_id": null,
        "id": "1193142949320474624",
        "qualified_id": null,
        "from_public": false,
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "create_time": "2024-01-06T10:44:07+08:00",
        "update_time": "2024-01-26T11:16:24+08:00",
        "self_defined_fields": [ ],
        "children_num": null,
        "children": null
      } ]
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码: 401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码: 403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

**状态码: 404**

Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchSubjectNewSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
```



```
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
SearchSubjectNewRequest request = new SearchSubjectNewRequest();
try {
    SearchSubjectNewResponse response = client.searchSubjectNew(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchSubjectNewRequest()
        response = client.search_subject_new(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchSubjectNewRequest{}
    response, err := client.SearchSubjectNew(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是CatalogVO数组, 以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

### 8.9.7 删除主题(新)

#### 功能介绍

删除主题(新)。

#### 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/{project\_id}/design/subjects

表 8-517 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-518 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-519 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。

参数	是否必选	参数类型	描述
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-520 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-521 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-522 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-523 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-524 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

表 8-525 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

删除主题(新)。

```
DELETE https://{endpoint}/v2/{project_id}/design/subjects
{
  "ids" : [ "1231174343611035648" ]
}
```

## 响应示例

**状态码：200**

Success, 返回data数据是删除个数。

```
{
  "value" : 1
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

删除主题(新)。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteSubjectNewSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteSubjectNewRequest request = new DeleteSubjectNewRequest();
IdsParam body = new IdsParam();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1231174343611035648");
body.withIds(listbodyIds);
request.withBody(body);
try {
    DeleteSubjectNewResponse response = client.deleteSubjectNew(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

删除主题(新)。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSubjectNewRequest()
        listbody = [
            "1231174343611035648"
        ]
        request.body = IdsParam(
            ids=listbody
        )
        response = client.delete_subject_new(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

删除主题(新)。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSubjectNewRequest{}
    var listIdsbody = []string{
        "1231174343611035648",
    }
    request.Body = &model.IdsParam{
        Ids: listIdsbody,
    }
    response, err := client.DeleteSubjectNew(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。



## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.9.8 创建主题(新)

### 功能介绍

创建主题(新)。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/design/subjects

表 8-526 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-527 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取(响应消息头中X-Subject-Token的值) 使用Token认证时必选。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-528 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码。更新时必填，创建时可以为空，ID字符串。
name_ch	是	String	中文名称。
name_en	是	String	英文名称。
description	否	String	描述信息，业务对象必填。
alias	否	String	别名。
data_owner	否	String	数据owner部门。
data_owner_list	是	String	数据owner人员。拼接成数组格式：["user_1","user_2"]。
level	是	Integer	层级。取值范围1-7。
parent_id	否	String	上层主题ID，首层则为空，ID字符串。
self_defined_fields	否	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。主题的自定义属性。

表 8-529 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

## 响应参数

状态码：200

表 8-530 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-531 data

参数	参数类型	描述
value	<b>CatalogVO</b> object	CatalogVO信息。

表 8-532 CatalogVO

参数	参数类型	描述
name_ch	String	中文名称。
name_en	String	英文名称。
description	String	描述信息。
qualified_name	String	扩展名。
guid	String	guid，自动生成。
code	String	编码。
alias	String	别名。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理，只读。

参数	参数类型	描述
data_owner	String	数据所有者。
data_owner_list	String	数据所有者集合。
data_department	String	数据域。
path	String	路径信息。
level	Integer	层级信息。
ordinal	Integer	序号。
owner	String	责任人。
parent_id	String	父目录ID，没有则为根目录，ID字符串。
swap_order_id	String	同层排序，目标节点的ID，ID字符串。
id	String	主题ID，ID字符串。
qualified_id	String	认证ID，自动生成。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
children_num	Integer	拥有子流程的数量，不包括子流程的子流程，前端不传。
children	Array of <b>CatalogVO</b> objects	下层子目录，只读。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	属性自定义项。

表 8-533 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-534 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-535 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-536 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-537 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404



表 8-538 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

创建主题(新)。

POST https://{endpoint}/v2/{project\_id}/design/subjects

```
{
  "name_ch": "test_create_catalog_demo",
  "name_en": "test_create_catalog_demo",
  "description": "this is a demo.",
  "alias": "",
  "data_owner": "",
  "data_owner_list": "[\"test_uesr\"]",
  "self_defined_fields": [ ],
  "level": 1
}
```

## 响应示例

状态码: 200

Success, 返回data数据是entity详情。

```
{
  "data": {
    "value": {
      "name_ch": "test_create_catalog_demo",
      "name_en": "test_create_catalog_demo",
      "description": "this is a demo.",
      "qualified_name": "test_create_catalog_demo@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "guid": null,
      "code": null,
      "status": "DRAFT",
      "new_biz": null,
      "alias": "",
      "data_owner": "",
      "data_owner_list": "[\"test_uesr\"]",
      "data_department": null,
      "path": "test_create_catalog_demo",
      "level": 1,
      "ordinal": 11,
      "owner": "test_uesr",
      "parent_id": null,
      "swap_order_id": null,
      "id": "1232018366517129216",
      "qualified_id": null,
      "from_public": null,
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "create_time": "2024-04-22T17:21:08.372+08:00",
      "update_time": "2024-04-22T17:21:08.372+08:00",
      "self_defined_fields": [ ],
      "children_num": null,
    }
  }
}
```

```
"children" : null
}
}
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建主题(新)。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateSubjectNewSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))
    .build();
CreateSubjectNewRequest request = new CreateSubjectNewRequest();
SubjectParamsVO body = new SubjectParamsVO();
body.withLevel(1);
body.withDataOwnerList("[\"test_uesr\"]");
body.withDataOwner("");
body.withAlias("");
body.withDescription("this is a demo.");
body.withNameEn("test_create_catalog_demo");
body.withNameCh("test_create_catalog_demo");
request.withBody(body);
try {
    CreateSubjectNewResponse response = client.createSubjectNew(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

创建主题(新)。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = CreateSubjectNewRequest()  
    request.body = SubjectParamsVO(  
        level=1,  
        data_owner_list=["test_uesr"],  
        data_owner="",  
        alias="",  
        description="this is a demo.",  
        name_en="test_create_catalog_demo",  
        name_ch="test_create_catalog_demo"  
    )  
    response = client.create_subject_new(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

## Go

创建主题(新)。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dataartsstudio.NewDataArtsStudioClient(  
        dataartsstudio.DataArtsStudioClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.CreateSubjectNewRequest{}  
    dataOwnerSubjectParamsVo:= ""  
    aliasSubjectParamsVo:= ""  
    descriptionSubjectParamsVo:= "this is a demo."  
    request.Body = &model.SubjectParamsVo{  
        Level: int32(1),  
        DataOwnerList: ["test_uesr"],  
        DataOwner: &dataOwnerSubjectParamsVo,  
        Alias: &aliasSubjectParamsVo,  
        Description: &descriptionSubjectParamsVo,  
    }
```

```
NameEn: "test_create_catalog_demo",  
NameCh: "test_create_catalog_demo",  
}  
response, err := client.CreateSubjectNew(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是entity详情。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.9.9 修改主题(新)

### 功能介绍

修改主题(新)。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/design/subjects

表 8-539 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-540 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-541 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码。更新时必填，创建时可以为空，ID字符串。
name_ch	是	String	中文名称。
name_en	是	String	英文名称。
description	否	String	描述信息，业务对象必填。
alias	否	String	别名。
data_owner	否	String	数据owner部门。
data_owner_list	是	String	数据owner人员。拼接成数组格式：["user_1","user_2"]。
level	是	Integer	层级。取值范围1-7。
parent_id	否	String	上层主题ID，首层则为空，ID字符串。
self_defined_fields	否	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。主题的自定义属性。

表 8-542 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

## 响应参数

状态码：200

表 8-543 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-544 data

参数	参数类型	描述
value	<b>CatalogVO</b> object	CatalogVO信息。

表 8-545 CatalogVO

参数	参数类型	描述
name_ch	String	中文名称。
name_en	String	英文名称。
description	String	描述信息。
qualified_name	String	扩展名。
guid	String	guid，自动生成。
code	String	编码。
alias	String	别名。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。
data_owner	String	数据所有者。
data_owner_list	String	数据所有者集合。
data_department	String	数据域。
path	String	路径信息。
level	Integer	层级信息。
ordinal	Integer	序号。
owner	String	责任人。
parent_id	String	父目录ID，没有则为根目录，ID字符串。
swap_order_id	String	同层排序，目标节点的ID，ID字符串。
id	String	主题ID，ID字符串。
qualified_id	String	认证ID，自动生成。
from_public	Boolean	是否来自公共层。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



参数	参数类型	描述
children_num	Integer	拥有子流程的数量，不包括子流程的子流程，前端不传。
children	Array of <b>CatalogVO</b> objects	下层子目录，只读。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	属性自定义项。

表 8-546 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-547 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-548 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-549 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-550 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：404

表 8-551 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

修改主题(新)。

PUT https://{endpoint}/v2/{project\_id}/design/subjects

```
{
  "id": "1231174343611035648",
  "name_ch": "test_update_catalog",
  "name_en": "test_create_catalog",
  "description": "this is a update demo.",
  "alias": "",
  "data_owner": "dataOwner",
  "data_owner_list": ["test_uesr"],
  "self_defined_fields": [ ],
  "level": 1
}
```

## 响应示例

**状态码：200**

Success，返回data数据是CatalogVO详情。

```
{
  "data": {
    "value": {
      "name_ch": "test_update_catalog",
      "name_en": "test_create_catalog",
      "description": "this is a update demo.",
      "qualified_name": "test_create_catalog@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
      "guid": null,
      "code": null,
      "status": "DRAFT",
      "new_biz": null,
      "alias": "",
      "data_owner": "dataOwner",
      "data_owner_list": ["test_uesr"],
      "data_department": null,
      "path": "test_update_catalog",
      "level": 1,
      "ordinal": null,
      "owner": "test_uesr",
      "parent_id": null,
      "swap_order_id": null,
      "id": "1231174343611035648",
      "qualified_id": null,
      "from_public": null,
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "create_time": "2024-04-20T09:27:18+08:00",
      "update_time": "2024-04-22T17:36:23.829+08:00",
      "self_defined_fields": [ ],
    }
  }
}
```

```
"children_num" : null,  
  "children" : null  
}  
}  
}
```

#### 状态码：400

BadRequest

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user request is illegal."  
}
```

#### 状态码：401

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

#### 状态码：403

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

#### 状态码：404

Not Found

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The User Request API does not exist."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

修改主题(新)。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class UpdateSubjectNewSolution {  
  
    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateSubjectNewRequest request = new UpdateSubjectNewRequest();
SubjectParamsVO body = new SubjectParamsVO();
body.withLevel(1);
body.withDataOwnerList("[\"test_uesr\"]");
body.withDataOwner("dataOwner");
body.withAlias("");
body.withDescription("this is a update demo.");
body.withNameEn("test_create_catalog");
body.withNameCh("test_update_catalog");
body.withId("1231174343611035648");
request.withBody(body);
try {
    UpdateSubjectNewResponse response = client.updateSubjectNew(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

修改主题(新)。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateSubjectNewRequest()
    request.body = SubjectParamsVO(
        level=1,
        data_owner_list=["test_uesr"],
        data_owner="dataOwner",
        alias="",
        description="this is a update demo.",
        name_en="test_create_catalog",
        name_ch="test_update_catalog",
        id="1231174343611035648"
    )
    response = client.update_subject_new(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

修改主题(新)。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSubjectNewRequest{}
    dataOwnerSubjectParamsVo := "dataOwner"
    aliasSubjectParamsVo := ""
    descriptionSubjectParamsVo := "this is a update demo."
    idSubjectParamsVo := "1231174343611035648"
    request.Body = &model.SubjectParamsVo{
        Level: int32(1),
```



```

DataOwnerList: "["test_uesr"]",
DataOwner: &dataOwnerSubjectParamsVo,
Alias: &aliasSubjectParamsVo,
Description: &descriptionSubjectParamsVo,
NameEn: "test_create_catalog",
NameCh: "test_update_catalog",
Id: &idSubjectParamsVo,
}
response, err := client.UpdateSubjectNew(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是CatalogVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.10 主题层级接口

### 8.10.1 获取主题层级

#### 功能介绍

获取主题层级。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/subject-levels

表 8-552 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-553 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-554 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-555 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-556 data

参数	参数类型	描述
value	Array of <b>CatalogLevelVO</b> objects	value，统一的返回结果的外层数据结构。

表 8-557 CatalogLevelVO

参数	参数类型	描述
id	String	编号，ID字符串。
level	Integer	层级。取值范围为1-7。
name_ch	String	中文名称。
name_en	String	英文名称。

状态码：400

表 8-558 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-559 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

**状态码：403**

**表 8-560 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

**表 8-561 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

获取当前工作空间下，主题的层级信息。

GET https://{endpoint}/v2/{project\_id}/design/subject-levels

## 响应示例

**状态码：200**

Success，返回层级列表。

```
{
  "data": {
    "value": [ {
      "id": "1141755876370214912",
      "level": 7,
      "name_ch": "业务对象",
      "name_en": "Business Object"
    }, {
      "id": "1141755876366020608",
      "level": 2,
      "name_ch": "主题域",
      "name_en": "Business Domain"
    }, {

```

```
{
  "id": "1184953097022242816",
  "level": 3,
  "name_ch": "主题域L3",
  "name_en": "Business Domain L3"
}, {
  "id": "1184953097022242817",
  "level": 4,
  "name_ch": "主题域L4",
  "name_en": "Business Domain L4"
}, {
  "id": "1184953097022242818",
  "level": 5,
  "name_ch": "主题域L5",
  "name_en": "Business Domain L5"
}, {
  "id": "1184953097022242819",
  "level": 6,
  "name_ch": "主题域L6",
  "name_en": "Business Domain L6"
}, {
  "id": "1141755876357632000",
  "level": 1,
  "name_ch": "主题域分组",
  "name_en": "Business Domain Group"
}
}]
}
```

#### 状态码: 400

##### BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

#### 状态码: 401

##### Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

#### 状态码: 403

##### Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

#### 状态码: 404

##### Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSubjectLevelsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSubjectLevelsRequest request = new ListSubjectLevelsRequest();
        try {
            ListSubjectLevelsResponse response = client.listSubjectLevels(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSubjectLevelsRequest()
    response = client.list_subject_levels(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSubjectLevelsRequest{}
    response, err := client.ListSubjectLevels(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回层级列表。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.10.2 修改或删除主题层级

### 功能介绍

修改或删除主题层级。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/subject-levels

表 8-562 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-563 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 (响应消息头中X-Subject-Token的值) 使用Token认证时必选。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。



参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-564 请求 Body 参数

参数	是否必选	参数类型	描述
levels	否	Array of <a href="#">CatalogLevelVO</a> objects	主题层级信息。

表 8-565 CatalogLevelVO

参数	是否必选	参数类型	描述
id	否	String	编号，ID字符串。
level	否	Integer	层级。取值范围为1-7。
name_ch	否	String	中文名称。
name_en	否	String	英文名称。

## 响应参数

状态码：200

表 8-566 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-567 data

参数	参数类型	描述
value	CatalogLevelVOList object	value, 统一的返回结果的外层数据结构。

表 8-568 CatalogLevelVOList

参数	参数类型	描述
levels	Array of CatalogLevelVO objects	主题层级信息。

表 8-569 CatalogLevelVO

参数	参数类型	描述
id	String	编号, ID字符串。
level	Integer	层级。取值范围为1-7。
name_ch	String	中文名称。
name_en	String	英文名称。

状态码: 400

表 8-570 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码: 401

表 8-571 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

**状态码：403**

表 8-572 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：404**

表 8-573 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

获取当前工作空间下，主题的层级信息。

PUT https://{endpoint}/v2/{project\_id}/design/subject-levels

```
{
  "levels": [ {
    "id": "1141755876357632000",
    "level": 1,
    "name_ch": "主题域分组",
    "name_en": "Business Domain Group"
  }, {
    "id": "1141755876366020608",
    "level": 2,
    "name_ch": "主题域",
    "name_en": "Business Domain"
  }, {
    "id": "1184953097022242816",
    "level": 3,
    "name_ch": "主题域L3",
    "name_en": "Business Domain L3"
  }, {
    "id": "1184953097022242817",
    "level": 4,
    "name_ch": "主题域L4",
    "name_en": "Business Domain L4"
  }
]
```

```
}, {  
  "id": "1184953097022242818",  
  "level": 5,  
  "name_ch": "主题域L5",  
  "name_en": "Business Domain L5"  
}, {  
  "id": "1141755876370214912",  
  "level": 6,  
  "name_ch": "业务对象",  
  "name_en": "Business Object"  
}]  
}
```

## 响应示例

### 状态码：200

Success, 返回数据是CatalogLevelVOList。

```
{  
  "data": {  
    "value": {  
      "levels": [ {  
        "id": "1141755876357632000",  
        "level": 1,  
        "name_ch": "主题域分组",  
        "name_en": "Business Domain Group"  
      }, {  
        "id": "1141755876366020608",  
        "level": 2,  
        "name_ch": "主题域",  
        "name_en": "Business Domain"  
      }, {  
        "id": "1184953097022242816",  
        "level": 3,  
        "name_ch": "主题域L3",  
        "name_en": "Business Domain L3"  
      }, {  
        "id": "1184953097022242817",  
        "level": 4,  
        "name_ch": "主题域L4",  
        "name_en": "Business Domain L4"  
      }, {  
        "id": "1184953097022242818",  
        "level": 5,  
        "name_ch": "主题域L5",  
        "name_en": "Business Domain L5"  
      }, {  
        "id": "1141755876370214912",  
        "level": 6,  
        "name_ch": "业务对象",  
        "name_en": "Business Object"  
      }  
    ]  
  }  
}
```

### 状态码：400

BadRequest

```
{  
  "error_code": "DS.60xx",  
  "error_msg": "The user request is illegal."  
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

**状态码：404**

Not Found

```
{
  "error_code": "DS.60xx",
  "error_msg": "The User Request API does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

获取当前工作空间下，主题的层级信息。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ChangeSubjectsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ChangeSubjectsRequest request = new ChangeSubjectsRequest();
```

```
CatalogLevelVOList body = new CatalogLevelVOList();
List<CatalogLevelVO> listbodyLevels = new ArrayList<>();
listbodyLevels.add(
    new CatalogLevelVO()
        .withId("1141755876357632000")
        .withLevel(1)
        .withNameCh("主题域分组")
        .withNameEn("Business Domain Group")
);
listbodyLevels.add(
    new CatalogLevelVO()
        .withId("1141755876366020608")
        .withLevel(2)
        .withNameCh("主题域")
        .withNameEn("Business Domain")
);
listbodyLevels.add(
    new CatalogLevelVO()
        .withId("1184953097022242816")
        .withLevel(3)
        .withNameCh("主题域L3")
        .withNameEn("Business Domain L3")
);
listbodyLevels.add(
    new CatalogLevelVO()
        .withId("1184953097022242817")
        .withLevel(4)
        .withNameCh("主题域L4")
        .withNameEn("Business Domain L4")
);
listbodyLevels.add(
    new CatalogLevelVO()
        .withId("1184953097022242818")
        .withLevel(5)
        .withNameCh("主题域L5")
        .withNameEn("Business Domain L5")
);
listbodyLevels.add(
    new CatalogLevelVO()
        .withId("1141755876370214912")
        .withLevel(6)
        .withNameCh("业务对象")
        .withNameEn("Business Object")
);
body.withLevels(listbodyLevels);
request.withBody(body);
try {
    ChangeSubjectsResponse response = client.changeSubjects(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

获取当前工作空间下，主题的层级信息。

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ChangeSubjectsRequest()
        listLevelsbody = [
            CatalogLevelVO(
                id="1141755876357632000",
                level=1,
                name_ch="主题域分组",
                name_en="Business Domain Group"
            ),
            CatalogLevelVO(
                id="1141755876366020608",
                level=2,
                name_ch="主题域",
                name_en="Business Domain"
            ),
            CatalogLevelVO(
                id="1184953097022242816",
                level=3,
                name_ch="主题域L3",
                name_en="Business Domain L3"
            ),
            CatalogLevelVO(
                id="1184953097022242817",
                level=4,
                name_ch="主题域L4",
                name_en="Business Domain L4"
            ),
            CatalogLevelVO(
                id="1184953097022242818",
                level=5,
                name_ch="主题域L5",
                name_en="Business Domain L5"
            ),
            CatalogLevelVO(
                id="1141755876370214912",
                level=6,
                name_ch="业务对象",
                name_en="Business Object"
            )
        ]
        request.body = CatalogLevelVOList(
            levels=listLevelsbody
        )
        response = client.change_subjects(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

获取当前工作空间下，主题的层级信息。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeSubjectsRequest{}
    idLevels:= "1141755876357632000"
    levelLevels:= int32(1)
    nameChLevels:= "主题域分组"
    nameEnLevels:= "Business Domain Group"
    idLevels1:= "1141755876366020608"
    levelLevels1:= int32(2)
    nameChLevels1:= "主题域"
    nameEnLevels1:= "Business Domain"
    idLevels2:= "1184953097022242816"
    levelLevels2:= int32(3)
    nameChLevels2:= "主题域L3"
    nameEnLevels2:= "Business Domain L3"
    idLevels3:= "1184953097022242817"
    levelLevels3:= int32(4)
    nameChLevels3:= "主题域L4"
    nameEnLevels3:= "Business Domain L4"
    idLevels4:= "1184953097022242818"
    levelLevels4:= int32(5)
    nameChLevels4:= "主题域L5"
    nameEnLevels4:= "Business Domain L5"
    idLevels5:= "1141755876370214912"
    levelLevels5:= int32(6)
    nameChLevels5:= "业务对象"
    nameEnLevels5:= "Business Object"
    var listLevelsbody = []model.CatalogLevelVo{
        {
            Id: &idLevels,
```



```

Level: &levelLevels,
NameCh: &nameChLevels,
NameEn: &nameEnLevels,
},
{
  Id: &idLevels1,
  Level: &levelLevels1,
  NameCh: &nameChLevels1,
  NameEn: &nameEnLevels1,
},
{
  Id: &idLevels2,
  Level: &levelLevels2,
  NameCh: &nameChLevels2,
  NameEn: &nameEnLevels2,
},
{
  Id: &idLevels3,
  Level: &levelLevels3,
  NameCh: &nameChLevels3,
  NameEn: &nameEnLevels3,
},
{
  Id: &idLevels4,
  Level: &levelLevels4,
  NameCh: &nameChLevels4,
  NameEn: &nameEnLevels4,
},
{
  Id: &idLevels5,
  Level: &levelLevels5,
  NameCh: &nameChLevels5,
  NameEn: &nameEnLevels5,
},
}
}
request.Body = &model.CatalogLevelVoList{
  Levels: &listLevelsbody,
}
response, err := client.ChangeSubjects(request)
if err == nil {
  fmt.Printf("%+v\n", response)
} else {
  fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回数据是CatalogLevelVoList。
400	BadRequest
401	Unauthorized
403	Forbidden
404	Not Found

## 8.11 目录管理

### 8.11.1 获取所有目录

#### 功能介绍

获取所有目录(数据标准、码表)。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/directories

表 8-574 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-575 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数, 即查询Y条数据。默认值50, 取值范围[1,100]。
offset	否	Integer	查询起始坐标, 即跳过X条数据, 仅支持0或limit的整数倍, 不满足则向下取整, 默认值0。
type	是	String	获取该目录下的数据, 如果有子目录, 获取所有子目录的数据。 枚举值: <ul style="list-style-type: none"> <li>• CODE: 码表目录</li> <li>• STANDARD_ELEMENT: 数据标准目录</li> </ul>

## 请求参数

表 8-576 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-577 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-578 data

参数	参数类型	描述
value	Array of <a href="#">DirectoryVO</a> objects	value，统一的返回结果的外层数据结构。

表 8-579 DirectoryVO

参数	参数类型	描述
name	String	目录名称。
description	String	描述。
type	String	目录类型。 枚举值： <ul style="list-style-type: none"> <li>STANDARD_ELEMENT: 数据标准</li> <li>CODE: 码表</li> </ul>
id	String	ID，创建时可不传，更新时必须填。ID字符串。
parent_id	String	父目录ID，首层传null。ID字符串。
prev_id	String	上个节点ID，首节点传null。ID字符串。
root_id	String	根节点ID，根节点此ID为自身ID，只读。ID字符串。
qualified_name	String	目录的资产名称，只读。
from_public	Boolean	是否来自公共层，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
ref_id	String	关联的主题ID，ID字符串。
children	Array of <a href="#">DirectoryVO</a> objects	子目录。

状态码：400

表 8-580 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-581 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-582 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据目录类型获取数据标准的全部目录。

```
GET https://{endpoint}/v2/{project_id}/design/directories?type=STANDARD_ELEMENT
```

## 响应示例

**状态码：200**

Success，返回data数据是带有子目录的目录。

```
{
  "data": {
    "value": [ {
      "name": "付款方式",
      "description": null,
      "type": "STANDARD_ELEMENT",
      "id": "1169318103583268864",
      "parent_id": null,
      "prev_id": null,
      "root_id": "1169318103583268864",
      "qualified_name": "付款方式",
    }
  ]
}
```

```
"create_time" : "2023-11-01T16:52:41+08:00",
"update_time" : "2023-11-01T16:52:41+08:00",
"from_public" : false,
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"ref_id" : null,
"children" : [ {
  "name" : "线上支付",
  "description" : null,
  "type" : "STANDARD_ELEMENT",
  "id" : "1231215227505364992",
  "parent_id" : "1169318103583268864",
  "prev_id" : null,
  "root_id" : "1169318103583268864",
  "qualified_name" : "付款方式.线上支付",
  "create_time" : "2024-04-20T12:09:45+08:00",
  "update_time" : "2024-04-20T12:09:45+08:00",
  "from_public" : false,
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "ref_id" : null,
  "children" : [ ]
} ]
} ]
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDirectoriesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDirectoriesRequest request = new ListDirectoriesRequest();
        try {
            ListDirectoriesResponse response = client.listDirectories(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = ListDirectoriesRequest()
    response = client.list_directories(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDirectoriesRequest{}
    response, err := client.ListDirectories(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是带有子目录的目录。



状态码	描述
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.11.2 创建目录

### 功能介绍

创建目录（数据标准、码表）。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/directories

表 8-583 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-584 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

表 8-585 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	目录名称。
description	否	String	描述。
type	是	String	目录类型。 枚举值: <ul style="list-style-type: none"> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• CODE: 码表</li> </ul>
id	否	String	ID, 创建时可不传, 更新时必须填。ID字符串。
parent_id	是	String	父目录ID, 首层传null。ID字符串。
prev_id	是	String	上个节点ID, 首节点传null。ID字符串。
ref_id	否	String	关联的主题ID, ID字符串。
children	否	Array of <b>DirectoryVO</b> objects	子目录。

## 响应参数

状态码: 200

表 8-586 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-587 data

参数	参数类型	描述
value	<b>DirectoryVO</b> object	value，统一的返回结果的外层数据结构。

表 8-588 DirectoryVO

参数	参数类型	描述
name	String	目录名称。
description	String	描述。
type	String	目录类型。 枚举值： <ul style="list-style-type: none"> <li>STANDARD_ELEMENT: 数据标准</li> <li>CODE: 码表</li> </ul>
id	String	ID，创建时可不传，更新时必须填。ID字符串。
parent_id	String	父目录ID，首层传null。ID字符串。
prev_id	String	上个节点ID，首节点传null。ID字符串。
root_id	String	根节点ID，根节点此ID为自身ID，只读。ID字符串。
qualified_name	String	目录的资产名称，只读。
from_public	Boolean	是否来自公共层，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
ref_id	String	关联的主题ID，ID字符串。
children	Array of <b>DirectoryVO</b> objects	子目录。

**状态码：400**

**表 8-589** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-590** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-591** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，创建目录。

```
POST https://{endpoint}/v2/{project_id}/design/directories?type=STANDARD_ELEMENT
{
  "name": "线上支付",
  "parent_id": "1169318103583268864",
  "prev_id": "1169318103583268864",
  "type": "STANDARD_ELEMENT"
}
```

## 响应示例

**状态码：200**

Success, 返回数据是DirectoryVO。

```
{
  "data": {
    "value": {
      "name": "线上支付",
      "description": null,
      "type": "STANDARD_ELEMENT",
      "id": "1231215227505364992",
      "parent_id": "1169318103583268864",
      "prev_id": null,
      "root_id": "1169318103583268864",
      "qualified_name": "付款方式.线上支付",
      "create_time": "2024-04-20T12:09:45.11+08:00",
      "update_time": "2024-04-20T12:09:45.11+08:00",
      "from_public": null,
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "ref_id": null,
      "children": null
    }
  }
}
```

**状态码: 400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码: 401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码: 403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，创建目录。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateDirectorySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateDirectoryRequest request = new CreateDirectoryRequest();
        DirectoryVO body = new DirectoryVO();
        body.withPrevId("1169318103583268864");
        body.withParentId("1169318103583268864");
        body.withType(DirectoryVO.TypeEnum.fromValue("STANDARD_ELEMENT"));
        body.withName("线上支付");
        request.withBody(body);
        try {
            CreateDirectoryResponse response = client.createDirectory(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据请求参数，创建目录。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
```

```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateDirectoryRequest()
    request.body = DirectoryVO(
        prev_id="1169318103583268864",
        parent_id="1169318103583268864",
        type="STANDARD_ELEMENT",
        name="线上支付"
    )
    response = client.create_directory(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据请求参数，创建目录。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDirectoryRequest{}
    request.Body = &model.DirectoryVo{
        Previd: "1169318103583268864",
        ParentId: "1169318103583268864",
        Type: model.GetDirectoryVoTypeEnum().STANDARD_ELEMENT,
        Name: "线上支付",
    }
    response, err := client.CreateDirectory(request)
```

```
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回数据是DirectoryVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.11.3 修改目录

### 功能介绍

修改目录（数据标准、码表）。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/directories

表 8-592 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。



## 请求参数

表 8-593 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-594 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	目录名称。
description	否	String	描述。
type	是	String	目录类型。 枚举值： <ul style="list-style-type: none"> <li>STANDARD_ELEMENT: 数据标准</li> <li>CODE: 码表</li> </ul>
id	否	String	ID，创建时可不传，更新时必须填。ID字符串。
parent_id	是	String	父目录ID，首层传null。ID字符串。
prev_id	是	String	上个节点ID，首节点传null。ID字符串。
ref_id	否	String	关联的主题ID，ID字符串。

参数	是否必选	参数类型	描述
children	否	Array of <b>DirectoryVO</b> objects	子目录。

## 响应参数

状态码：200

表 8-595 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-596 data

参数	参数类型	描述
value	<b>DirectoryVO</b> object	value，统一的返回结果的外层数据结构。

表 8-597 DirectoryVO

参数	参数类型	描述
name	String	目录名称。
description	String	描述。
type	String	目录类型。 枚举值： <ul style="list-style-type: none"> <li>STANDARD_ELEMENT: 数据标准</li> <li>CODE: 码表</li> </ul>
id	String	ID，创建时可不传，更新时必须填。ID字符串。
parent_id	String	父目录ID，首层传null。ID字符串。
prev_id	String	上个节点ID，首节点传null。ID字符串。
root_id	String	根节点ID，根节点此ID为自身ID，只读。ID字符串。
qualified_name	String	目录的资产名称，只读。

参数	参数类型	描述
from_public	Boolean	是否来自公共层，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
ref_id	String	关联的主题ID，ID字符串。
children	Array of <b>DirectoryVO</b> objects	子目录。

**状态码：400**

表 8-598 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-599 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-600 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，修改目录。

```
PUT https://{endpoint}/v2/{project_id}/design/directories?type=STANDARD_ELEMENT
{
  "name": "线上支付",
  "parent_id": "1169318103583268864",
  "prev_id": "1169318103583268864",
  "type": "STANDARD_ELEMENT",
  "id": "1231215227505364992"
}
```

## 响应示例

**状态码：200**

Success，返回数据是DirectoryVO。

```
{
  "data": {
    "value": {
      "name": "线上支付",
      "description": null,
      "type": "STANDARD_ELEMENT",
      "id": "1231215227505364992",
      "parent_id": "1169318103583268864",
      "prev_id": null,
      "root_id": "1169318103583268864",
      "qualified_name": null,
      "create_time": null,
      "update_time": null,
      "from_public": null,
      "create_by": null,
      "update_by": "0833a573fe80d5401f6dc010a775569a",
      "ref_id": null,
      "children": null
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，修改目录。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateDirectorySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateDirectoryRequest request = new UpdateDirectoryRequest();
        DirectoryVO body = new DirectoryVO();
        body.withPrevId("1169318103583268864");
        body.withParentId("1169318103583268864");
        body.withId("1231215227505364992");
        body.withType(DirectoryVO.TypeEnum.fromValue("STANDARD_ELEMENT"));
        body.withName("线上支付");
        request.withBody(body);
        try {
            UpdateDirectoryResponse response = client.updateDirectory(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

根据请求参数，修改目录。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateDirectoryRequest()
        request.body = DirectoryVO(
            prev_id="1169318103583268864",
            parent_id="1169318103583268864",
            id="1231215227505364992",
            type="STANDARD_ELEMENT",
            name="线上支付"
        )
        response = client.update_directory(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据请求参数，修改目录。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```

dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDirectoryRequest{}
    idDirectoryVo:= "1231215227505364992"
    request.Body = &model.DirectoryVo{
        Previd: "1169318103583268864",
        ParentId: "1169318103583268864",
        Id: &idDirectoryVo,
        Type: model.GetDirectoryVoTypeEnum().STANDARD_ELEMENT,
        Name: "线上支付",
    }
    response, err := client.UpdateDirectory(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回数据是DirectoryVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.11.4 删除目录

### 功能介绍

删除目录（数据标准、码表）。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/directories

表 8-601 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-602 Query 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	实体ID数组，ID字符串。

### 请求参数

表 8-603 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。



参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-604 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果, 返回成功删除的对象个数。

表 8-605 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码: 400

表 8-606 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码: 401

表 8-607 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：403

表 8-608 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，修改目录。

```
DELETE https://{endpoint}/v2/{project_id}/design/directories?ids=1231215227505364992
```

## 响应示例

状态码：400

BadRequest

```
{  
  "error_code": "DS.60xx",  
  "error_msg": "The user request is illegal."  
}
```

状态码：401

Unauthorized

```
{  
  "error_code": "DS.60xx",  
  "error_msg": "User authentication failed."  
}
```

状态码：403

Forbidden

```
{  
  "error_code": "DS.60xx",  
  "error_msg": "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteDirectorySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteDirectoryRequest request = new DeleteDirectoryRequest();
        try {
            DeleteDirectoryResponse response = client.deleteDirectory(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteDirectoryRequest()
    response = client.delete_directory(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDirectoryRequest{}
    response, err := client.DeleteDirectory(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.12 原子指标接口

### 8.12.1 查找原子指标

#### 功能介绍

通过中英文名称、创建者、审核人、状态、修改时间分页查找原子指标信息看，中英文名称支持模糊查询。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/atomic-indexs

表 8-609 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-610 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。

参数	是否必选	参数类型	描述
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l3_id	否	String	业务对象l3的ID，ID字符串。
table_id	否	String	关联表的ID，ID字符串。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-611 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-612 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	返回数据。

表 8-613 data

参数	参数类型	描述
value	<a href="#">value</a> object	返回的数据信息。

表 8-614 value

参数	参数类型	描述
records	Array of <b>AtomicIndexVO</b> objects	AtomicIndexVO数组。
total	Integer	总数。

表 8-615 AtomicIndexVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	原子指标英文名。
name_ch	String	原子指标英文名。
description	String	描述。
create_by	String	创建人。
cal_exp	String	计算表达式，形如'sum(\${fact_column_id})'，其中fact_column_id表示引用事实表中的字段ID。
cal_fn_ids	Array of strings	引用函数ID，ID字符串。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象guid，ID字符串。
table_id	String	事实表ID，ID字符串。
tb_name	String	事实表名称。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
field_ids	Array of strings	字段ID信息，ID字符串。
field_names	Array of strings	字段名称信息。



参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"><li>• DRAFT: 草稿</li><li>• PUBLISH_DEVELOPING: 发布待审核</li><li>• PUBLISHED: 已发布</li><li>• OFFLINE_DEVELOPING: 下线待审核</li><li>• OFFLINE: 已下线</li><li>• REJECT: 已驳回</li></ul>

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。

表 8-616 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-617 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-618 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-619 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-620 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询中文名中包含‘test\_a’的原子指标集合，查询前十条。

```
GET https://{endpoint}/v2/{project_id}/design/atomic-indexs?offset=1&limit=10&name=test_a
```

## 响应示例

状态码：200

Success，返回data数据是AtomicIndexVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1194663814218559488",
        "name_en": "test_atomic",
```

```
"name_ch": "test_atomic",
"description": "",
"create_by": "test_uesr",
"cal_exp": "avg(${1184822297161261056}) ",
"cal_fn_ids": null,
"l1_id": "1080855591154495488",
"l2_id": null,
"l3_id": null,
"table_id": "1184822297001877504",
"tb_name": "test_111",
"dw_type": "DWS",
"field_ids": [ "1184822297161261056" ],
"field_names": null,
"status": "PUBLISHED",
"biz_type": "FACT_LOGIC_TABLE",
"create_time": "2024-01-10T15:27:29+08:00",
"update_time": "2024-01-10T15:29:30+08:00",
"l1": "ceshi_test_2",
"l2": null,
"l3": null,
"approval_info": {
  "id": "1194664324778602496",
  "tenant_id": null,
  "name_ch": null,
  "name_en": null,
  "biz_id": "1194663814218559488",
  "biz_type": null,
  "biz_info": null,
  "biz_info_obj": null,
  "biz_version": 0,
  "biz_status": "PUBLISHED",
  "approval_status": "APPROVED",
  "approval_type": null,
  "submit_time": "2024-01-10T15:29:31+08:00",
  "create_by": null,
  "l1": null,
  "l2": null,
  "l3": null,
  "approval_time": "2024-01-10T15:29:31+08:00",
  "approver": "test_uesr",
  "email": null,
  "msg": "Fast Approval OK. (Associated Publish By test_atomic, Automatic Approval.)",
  "directory_path": null
},
"new_biz": null
}]
}
}
```

#### 状态码: 400

##### BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

#### 状态码: 401

##### Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

#### 状态码: 403

## Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchAtomicIndexesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        SearchAtomicIndexesRequest request = new SearchAtomicIndexesRequest();
        try {
            SearchAtomicIndexesResponse response = client.searchAtomicIndexes(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchAtomicIndexesRequest()
        response = client.search_atomic_indexes(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchAtomicIndexesRequest{}
    response, err := client.SearchAtomicIndexes(request)
```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是AtomicIndexVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.12.2 新建原子指标

### 功能介绍

新建单个原子指标。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/atomic-indexs

表 8-621 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-622 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-623 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name_en	是	String	原子指标英文名。
name_ch	是	String	原子指标英文名。
description	否	String	描述。
create_by	否	String	创建人。
cal_exp	是	String	计算表达式，形如'sum(\${fact_column_id})'，其中fact_column_id表示引用事实表中的字段ID。
cal_fn_ids	否	Array of strings	引用函数ID，ID字符串。
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。
l3_id	是	String	业务对象guid，ID字符串。
table_id	是	String	事实表ID，ID字符串。

参数	是否必选	参数类型	描述
tb_name	否	String	事实表名称。
field_ids	是	Array of strings	字段ID信息，ID字符串。
field_names	否	Array of strings	字段名称信息。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>● DRAFT: 草稿</li> <li>● PUBLISH_DEVELOPING: 发布待审核</li> <li>● PUBLISHED: 已发布</li> <li>● OFFLINE_DEVELOPING: 下线待审核</li> <li>● OFFLINE: 已下线</li> <li>● REJECT: 已驳回</li> </ul>



参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManagementVO object	业务版本管理，只读。

表 8-624 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID, ID字符串。
tenant_id	否	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-625 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>



参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

## 响应参数

状态码：200

表 8-626 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回数据。

表 8-627 data

参数	参数类型	描述
value	<b>AtomicIndexVO</b> object	返回的数据信息。

表 8-628 AtomicIndexVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	原子指标英文名。
name_ch	String	原子指标英文名。
description	String	描述。
create_by	String	创建人。

参数	参数类型	描述
cal_exp	String	计算表达式, 形如'sum({fact_column_id})', 其中 fact_column_id表示引用事实表中的字段 ID。
cal_fn_ids	Array of strings	引用函数ID, ID字符串。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象guid, ID字符串。
table_id	String	事实表ID, ID字符串。
tb_name	String	事实表名称。
dw_type	String	数据连接类型, 对应表所在的数仓类型, 取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
field_ids	Array of strings	字段ID信息, ID字符串。
field_names	Array of strings	字段名称信息。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。

表 8-629 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。



参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-630 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-631 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-632 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-633 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据下列参数，新建原子指标。

POST https://{endpoint}/v2/{project\_id}/design/atomic-indexs

```
{
  "id" : 0,
  "name_ch" : "test_create_atomic_index",
  "name_en" : "test_create_atomic_index",
  "table_id" : "1215358267694411776",
  "cal_exp" : "max(${1215358267774103552})",
  "l3_id" : "1080855591154495488",
  "field_ids" : [ "1215358267774103552" ],
  "biz_type" : "FACT_LOGIC_TABLE",
  "description" : "无"
}
```

## 响应示例

### 状态码：200

Success, 返回data数据是AtomicIndexVO。

```
{
  "data": {
    "value": {
      "id": "1227990857618227200",
      "name_en": "test_create_atomic_index",
      "name_ch": "test_create_atomic_index",
      "description": "无",
      "create_by": "test_uesr",
      "cal_exp": "max(${1215358267774103552})",
      "cal_fn_ids": null,
      "l1_id": null,
      "l2_id": null,
      "l3_id": "1080855591154495488",
      "table_id": "1215358267694411776",
      "tb_name": null,
      "dw_type": null,
      "field_ids": [ "1215358267774103552" ],
      "field_names": null,
      "status": "DRAFT",
      "biz_type": "FACT_LOGIC_TABLE",
      "create_time": "2024-04-11T14:37:15.423+08:00",
      "update_time": "2024-04-11T14:37:15.423+08:00",
      "l1": null,
      "l2": null,
      "l3": null,
      "approval_info": null,
      "new_biz": null
    }
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

根据下列参数，新建原子指标。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateDesignAtomicIndexSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateDesignAtomicIndexRequest request = new CreateDesignAtomicIndexRequest();
        AtomicIndexVO body = new AtomicIndexVO();
        List<String> listbodyFieldIds = new ArrayList<>();
        listbodyFieldIds.add("121535826774103552");
        body.withBizType(AtomicIndexVO.BizTypeEnum.fromValue("FACT_LOGIC_TABLE"));
        body.withFieldIds(listbodyFieldIds);
        body.withTableId("1215358267694411776");
        body.withL3Id("1080855591154495488");
        body.withCalExp("max({121535826774103552})");
        body.withDescription("无");
        body.withNameCh("test_create_atomic_index");
        body.withNameEn("test_create_atomic_index");
        body.withId("0");
        request.withBody(body);
        try {
            CreateDesignAtomicIndexResponse response = client.createDesignAtomicIndex(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据下列参数，新建原子指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateDesignAtomicIndexRequest()
        listFieldIdsbody = [
            "1215358267774103552"
        ]
        request.body = AtomicIndexVO(
            biz_type="FACT_LOGIC_TABLE",
            field_ids=listFieldIdsbody,
            table_id="1215358267694411776",
            l3_id="1080855591154495488",
            cal_exp="max(${1215358267774103552})",
            description="无",
            name_ch="test_create_atomic_index",
            name_en="test_create_atomic_index",
            id="0"
        )
        response = client.create_design_atomic_index(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据下列参数，新建原子指标。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
// risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
// variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
// example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateDesignAtomicIndexRequest{}
var listFieldIdsbody = []string{
    "1215358267774103552",
}
bizTypeBizType:= model.GetBizTypeEnumBizTypeEnum().FACT_LOGIC_TABLE
descriptionAtomicIndexVo:= "无"
idAtomicIndexVo:= "0"
request.Body = &model.AtomicIndexVo{
    BizType: &bizTypeBizType,
    FieldIds: listFieldIdsbody,
    TableId: "1215358267694411776",
    L3Id: "1080855591154495488",
    CalExp: "max(${1215358267774103552})",
    Description: &descriptionAtomicIndexVo,
    NameCh: "test_create_atomic_index",
    NameEn: "test_create_atomic_index",
    Id: &idAtomicIndexVo,
}
response, err := client.CreateDesignAtomicIndex(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是AtomicIndexVO。
400	BadRequest
401	Unauthorized
403	Forbidden



## 8.12.3 更新原子指标

### 功能介绍

更新单个原子指标。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/atomic-indexs

表 8-634 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-635 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-636 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name_en	是	String	原子指标英文名。
name_ch	是	String	原子指标英文名。
description	否	String	描述。
create_by	否	String	创建人。
cal_exp	是	String	计算表达式，形如'sum({fact_column_id})'，其中fact_column_id表示引用事实表中的字段ID。
cal_fn_ids	否	Array of strings	引用函数ID，ID字符串。
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。
l3_id	是	String	业务对象guid，ID字符串。
table_id	是	String	事实表ID，ID字符串。
tb_name	否	String	事实表名称。
field_ids	是	Array of strings	字段ID信息，ID字符串。
field_names	否	Array of strings	字段名称信息。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManagementVO object	业务版本管理，只读。

表 8-637 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。



表 8-638 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

## 响应参数

状态码：200

表 8-639 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回数据。

表 8-640 data

参数	参数类型	描述
value	<b>AtomicIndexVO</b> object	返回的数据信息。

表 8-641 AtomicIndexVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	原子指标英文名。
name_ch	String	原子指标英文名。
description	String	描述。
create_by	String	创建人。

参数	参数类型	描述
cal_exp	String	计算表达式, 形如'sum({fact_column_id})', 其中 fact_column_id表示引用事实表中的字段 ID。
cal_fn_ids	Array of strings	引用函数ID, ID字符串。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象guid, ID字符串。
table_id	String	事实表ID, ID字符串。
tb_name	String	事实表名称。
dw_type	String	数据连接类型, 对应表所在的数仓类型, 取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
field_ids	Array of strings	字段ID信息, ID字符串。
field_names	Array of strings	字段名称信息。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。

表 8-642 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-643 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-644 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-645 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-646 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据下列参数，更新原子指标。

PUT https://{endpoint}/v2/{project\_id}/design/atomic-indexs

```
{
  "id": "1227990857618227200",
  "name_ch": "test_update_atomic_index",
  "name_en": "test_update_atomic_index",
  "table_id": "1215358267694411776",
  "cal_exp": "max(${1215358267774103552})",
  "l3_id": "1080855591154495488",
  "field_ids": [ "1215358267774103552" ],
  "biz_type": "FACT_LOGIC_TABLE",
  "description": "这是原子指标的描述。"
}
```

## 响应示例

### 状态码：200

Success, 返回data数据是AtomicIndexVO。

```
{
  "data": {
    "value": {
      "id": "1227990857618227200",
      "name_en": "test_update_atomic_index",
      "name_ch": "test_update_atomic_index",
      "description": "这是原子指标的描述。",
      "create_by": "test_uesr",
      "cal_exp": "max(${1215358267774103552})",
      "cal_fn_ids": null,
      "l1_id": null,
      "l2_id": null,
      "l3_id": "1080855591154495488",
      "table_id": "1215358267694411776",
      "tb_name": null,
      "dw_type": null,
      "field_ids": [ "1215358267774103552" ],
      "field_names": null,
      "status": "DRAFT",
      "biz_type": "FACT_LOGIC_TABLE",
      "create_time": "2024-04-11T14:37:15+08:00",
      "update_time": "2024-04-11T14:45:26.179+08:00",
      "l1": null,
      "l2": null,
      "l3": null,
      "approval_info": null,
      "new_biz": null
    }
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

根据下列参数，更新原子指标。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateDesignAtomicIndexSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateDesignAtomicIndexRequest request = new UpdateDesignAtomicIndexRequest();
        AtomicIndexVO body = new AtomicIndexVO();
        List<String> listbodyFieldIds = new ArrayList<>();
        listbodyFieldIds.add("121535826774103552");
        body.withBizType(AtomicIndexVO.BizTypeEnum.fromValue("FACT_LOGIC_TABLE"));
        body.withFieldIds(listbodyFieldIds);
        body.withTableId("1215358267694411776");
        body.withL3Id("1080855591154495488");
        body.withCalExp("max({121535826774103552})");
        body.withDescription("这是原子指标的描述。");
        body.withNameCh("test_update_atomic_index");
        body.withNameEn("test_update_atomic_index");
        body.withId("1227990857618227200");
        request.withBody(body);
        try {
            UpdateDesignAtomicIndexResponse response = client.updateDesignAtomicIndex(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```



## Python

根据下列参数，更新原子指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateDesignAtomicIndexRequest()
        listFieldIdsbody = [
            "1215358267774103552"
        ]
        request.body = AtomicIndexVO(
            biz_type="FACT_LOGIC_TABLE",
            field_ids=listFieldIdsbody,
            table_id="1215358267694411776",
            l3_id="1080855591154495488",
            cal_exp="max({1215358267774103552})",
            description="这是原子指标的描述。",
            name_ch="test_update_atomic_index",
            name_en="test_update_atomic_index",
            id="1227990857618227200"
        )
        response = client.update_design_atomic_index(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据下列参数，更新原子指标。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
// risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
// variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
// example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateDesignAtomicIndexRequest{}
var listFieldIdsbody = []string{
    "1215358267774103552",
}
bizTypeBizType:= model.GetBizTypeEnumBizTypeEnum().FACT_LOGIC_TABLE
descriptionAtomicIndexVo:= "这是原子指标的描述。"
idAtomicIndexVo:= "1227990857618227200"
request.Body = &model.AtomicIndexVo{
    BizType: &bizTypeBizType,
    FieldIds: listFieldIdsbody,
    TableId: "1215358267694411776",
    L3Id: "1080855591154495488",
    CalExp: "max(${1215358267774103552})",
    Description: &descriptionAtomicIndexVo,
    NameCh: "test_update_atomic_index",
    NameEn: "test_update_atomic_index",
    Id: &idAtomicIndexVo,
}
response, err := client.UpdateDesignAtomicIndex(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是AtomicIndexVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.12.4 删除原子指标

### 功能介绍

批量删除原子指标。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/atomic-idxs

表 8-647 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-648 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-649 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表, ID字符串。
del_types	否	String	删除类型。 枚举值: <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表, 只针对能物化的表生效 (配置此值则表示删除)</li> </ul>

## 响应参数

状态码: 200

表 8-650 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果, 返回成功删除的对象个数。

表 8-651 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码: 400

表 8-652 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码: 401

表 8-653 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-654 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入的参数，删除原子指标，只支持删除状态为草稿、已下线、已驳回的原子指标。

```
DELETE https://{endpoint}/v2/{project_id}/design/atomic-idxs
{
  "ids" : [ "1227990857618227200" ]
}
```

## 响应示例

状态码：200

Success，返回data数据是删除个数。

```
{
  "value" : 1
}
```

状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
```

```
"error_msg" : "User authentication failed."  
}
```

**状态码：403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据传入的参数，删除原子指标，只支持删除状态为草稿、已下线、已驳回的原子指标。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class DeleteDesignAtomicIndexSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        DeleteDesignAtomicIndexRequest request = new DeleteDesignAtomicIndexRequest();  
        IdsParam body = new IdsParam();  
        List<String> listbodyIds = new ArrayList<>();  
        listbodyIds.add("1227990857618227200");  
        body.withIds(listbodyIds);  
        request.withBody(body);  
        try {  
            DeleteDesignAtomicIndexResponse response = client.deleteDesignAtomicIndex(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据传入的参数，删除原子指标，只支持删除状态为草稿、已下线、已驳回的原子指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteDesignAtomicIndexRequest()
        listIdsbody = [
            "1227990857618227200"
        ]
        request.body = IdsParam(
            ids=listIdsbody
        )
        response = client.delete_design_atomic_index(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据传入的参数，删除原子指标，只支持删除状态为草稿、已下线、已驳回的原子指标。

```
package main

import (
    "fmt"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDesignAtomicIndexRequest{}
    var listIdsbody = []string{
        "1227990857618227200",
    }
    request.Body = &model.IdsParam{
        Ids: listIdsbody,
    }
    response, err := client.DeleteDesignAtomicIndex(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden



## 8.12.5 查看原子指标详情

### 功能介绍

通过ID获取原子指标详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/atomic-indexs/{id}

表 8-655 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-656 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

### 请求参数

表 8-657 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-658 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回数据。

表 8-659 data

参数	参数类型	描述
value	<b>AtomicIndexVO</b> object	返回的数据信息。

表 8-660 AtomicIndexVO

参数	参数类型	描述
id	String	编码, ID字符串。
name_en	String	原子指标英文名。
name_ch	String	原子指标英文名。
description	String	描述。
create_by	String	创建人。
cal_exp	String	计算表达式, 形如'sum(\${fact_column_id})', 其中fact_column_id表示引用事实表中的字段ID。
cal_fn_ids	Array of strings	引用函数ID, ID字符串。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。

参数	参数类型	描述
l3_id	String	业务对象guid, ID字符串。
table_id	String	事实表ID, ID字符串。
tb_name	String	事实表名称。
dw_type	String	数据连接类型, 对应表所在的数仓类型, 取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
field_ids	Array of strings	字段ID信息, ID字符串。
field_names	Array of strings	字段名称信息。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。

表 8-661 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。



参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-662 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-663 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-664 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-665 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据原子指标ID，获取原子指标详情。

GET https://{endpoint}/v2/{project\_id}/design/atomic-indexs/1193149872312823808

## 响应示例

状态码：200

Success，返回data数据是AtomicIndexVO详情。

```
{
  "data": {
    "value": {
      "id": "1193149872312823808",
      "name_en": "countMoney",
      "name_ch": "计算总车费",
      "description": null,

```

```

"create_by" : "test_uesr",
"cal_exp" : "SUM({1169329693695123463})",
"cal_fn_ids" : null,
"l1_id" : "1169309252771299328",
"l2_id" : "1169309374670356480,1169309948774162432",
"l3_id" : null,
"table_id" : "1169329693619625984",
"tb_name" : "行程订单",
"dw_type" : "MRS_HIVE",
"field_ids" : [ "1169329693695123463" ],
"field_names" : [ "fare_amount" ],
"status" : "REJECT",
"biz_type" : "FACT_LOGIC_TABLE",
"create_time" : "2024-01-06T11:11:37+08:00",
"update_time" : "2024-01-22T20:30:44+08:00",
"l1" : "城市交通",
"l2" : "行程记录/标准记录",
"l3" : null,
"approval_info" : {
  "id" : "1193149901970747392",
  "tenant_id" : null,
  "name_ch" : null,
  "name_en" : null,
  "biz_id" : "1193149872312823808",
  "biz_type" : null,
  "biz_info" : null,
  "biz_info_obj" : null,
  "biz_version" : 0,
  "biz_status" : "REJECT",
  "approval_status" : "REJECT",
  "approval_type" : null,
  "submit_time" : "2024-01-06T11:11:44+08:00",
  "create_by" : null,
  "l1" : null,
  "l2" : null,
  "l3" : null,
  "approval_time" : "2024-01-22T20:30:44+08:00",
  "approver" : "test_uesr",
  "email" : null,
  "msg" : "no",
  "directory_path" : null
},
"new_biz" : null
}
}
}

```

**状态码: 400**

BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

**状态码: 401**

Unauthorized

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}

```

**状态码: 403**

Forbidden

```

{
  "error_code" : "DS.60xx",

```

```
"error_msg": "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowAtomicIndexByIdSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowAtomicIndexByIdRequest request = new ShowAtomicIndexByIdRequest();  
        request.withId("{id}");  
        try {  
            ShowAtomicIndexByIdResponse response = client.showAtomicIndexById(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowAtomicIndexByIdRequest()
        request.id = "{id}"
        response = client.show_atomic_index_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAtomicIndexByIdRequest{}
    request.Id = "{id}"
    response, err := client.ShowAtomicIndexById(request)
```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是AtomicIndexVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.13 衍生指标接口

### 8.13.1 查找衍生指标

#### 功能介绍

通过中英文名称、创建者、审核人、状态、修改时间、l3Id分页查找衍生指标信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/derivative-indexs

表 8-666 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。



表 8-667 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
dimension_id	否	String	依据维度ID查维度属性，ID字符串。
dimension_group	否	String	依据维度颗粒度查维度属性。
atomic_index_id	否	String	依据原子指标ID查维度属性，ID字符串。
all_metrics	否	Boolean	是否查询复合指标
dw_type	否	String	数据连接类型
l3_id	否	String	业务对象l3的ID，ID字符串。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-668 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-669 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	返回数据。

表 8-670 data

参数	参数类型	描述
value	value object	返回的数据信息。

表 8-671 value

参数	参数类型	描述
records	Array of <b>DerivativeIndexVO</b> objects	DerivativeIndexVO数组。
total	Integer	总数。

表 8-672 DerivativeIndexVO

参数	参数类型	描述
id	String	衍生指标ID, ID字符串。更新时必填。
name_en	String	衍生指标英文名称。
name_ch	String	衍生指标名称。
description	String	描述, 只读。
create_by	String	创建人。
data_type	String	字段类型。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象guid, ID字符串。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
atomic_index_id	String	原子指标ID, ID字符串。

参数	参数类型	描述
time_condition_id	String	时间限定ID, ID字符串。
time_field_id	String	时间限定关联字段ID, ID字符串。
time_field_name	String	时间限定关联字段名称, 只读。
common_conditions	Array of <b>CommonConditionVO</b> objects	通用限定信息。
dimension_groups	Array of <b>DerivativeIndexDimensionVO</b> objects	统计维度, 更新时不可修改。
monitor	<b>MetricMonitorVO</b> object	指标监控。
atomic_index	<b>AtomicIndexVO</b> object	原子指标信息, 名称、单位, 只读。
time_condition_name	String	时间限定名称, 只读。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名, 只读, 创建和更新时无需填写。
l2	String	主题域中文名, 只读, 创建和更新时无需填写。
l3	String	业务对象中文名, 只读, 创建和更新时无需填写。
summary_table_id	String	汇总表ID, 只读, ID字符串。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理, 只读。

表 8-673 CommonConditionVO

参数	参数类型	描述
name	String	名称。
field_ids	Array of strings	字段id信息，格式：table_id.field_id。
field_names	Array of strings	字段名称信息，格式：表名称.字段名称。
cal_exp	String	计算表达式，形如'\$ {table_id.column_id} > 1'，其中table_id表示引用字段所属表ID，column_id表示引用字段ID。
cal_fn_ids	Array of strings	引用函数ID，ID字符串。
front_configs	String	前端表达式配置，用于前端数据恢复。
id	String	通用限定ID，只读，ID字符串。

表 8-674 DerivativeIndexDimensionVO

参数	参数类型	描述
group_id	String	维度分组ID。
role	String	维度角色。
dimension_id	String	维度ID，ID字符串。
hierarchies_id	String	维度层级ID，ID字符串。
ordinal	Integer	序号，只读。
group_name	String	维度分组名称。
group_code	String	维度分组编码。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
hierarchies	Array of <a href="#">DimensionHierarchiesVO</a> objects	层级属性，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，只读，ID字符串。
dw_type	String	数据连接类型。
id	String	层级的ID，只读，ID字符串。

表 8-675 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	层级名称。

参数	参数类型	描述
attrs	Array of <b>HierarchiesAttrVO</b> objects	层级包含的属性。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-676 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码，ID字符串。
hierarchies_id	String	层级ID，ID字符串。
attr_id	String	属性ID，ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称，只读。
detail_attr_ids	Array of strings	详情属性ID，ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文，只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段，只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段，只读。



表 8-677 DimensionAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
dimension_id	String	维度ID，只读，ID字符串。
code_table_field_id	String	码表属性ID，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。

表 8-678 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-679 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理, 只读。

表 8-680 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-681 MetricMonitorVO

参数	参数类型	描述
id	String	编码, ID字符串。
other_metric_ids	Array of strings	其他指标ID, ID字符串。
other_metric_names	Array of strings	其他指标名称, 只读。

参数	参数类型	描述
other_compound_metric_ids	Array of strings	其他复合指标ID。
other_compound_metric_names	Array of strings	其他复合指标名称。
expression	String	告警表达式。
metric_id	String	挂载指ID, ID字符串。
front_configs	String	前端表达式配置, 用于前端数据恢复。

参数	参数类型	描述
metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

表 8-682 AtomicIndexVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	原子指标英文名。
name_ch	String	原子指标中文名。
description	String	描述。
create_by	String	创建人。
cal_exp	String	计算表达式，形如'sum(\${fact_column_id})'，其中fact_column_id表示引用事实表中的字段ID。
cal_fn_ids	Array of strings	引用函数ID，ID字符串。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象guid，ID字符串。
table_id	String	事实表ID，ID字符串。
tb_name	String	事实表名称。



参数	参数类型	描述
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
field_ids	Array of strings	字段ID信息，ID字符串。
field_names	Array of strings	字段名称信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。

表 8-683 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-684 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-685 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-686 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-687 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数参数，分页查询衍生指标。

```
GET https://{endpoint}/v2/{project_id}/design/derivative-idxs?offset=1&limit=10
```

## 响应示例

状态码：200

Success，返回data数据是DerivativeIndexVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1181167466668310528",
        "name_en": "sum_total_amount",

```

```

"name_ch": "总车费(供应商)",
"description": "",
"create_by": "test_uesr",
"data_type": "STRING",
"l1_id": "1170066794946420736",
"l2_id": null,
"l3_id": null,
"status": "PUBLISHED",
"atomic_index_id": "1170005876631875584",
"time_condition_id": null,
"time_field_id": null,
"time_field_name": null,
"common_conditions": [ ],
"dimension_groups": [ {
  "group_id": "1169322847009742848",
  "role": null,
  "dimension_id": "1169322847009742848",
  "hierarchies_id": null,
  "ordinal": 1,
  "group_name": "供应商",
  "group_code": "dim_vendor",
  "biz_type": "DIMENSION",
  "hierarchies": null,
  "l1": null,
  "l2": null,
  "l3": null,
  "l1_id": null,
  "l2_id": null,
  "l3_id": null,
  "dw_type": null,
  "id": "1181167466768973824"
}],
"monitor": null,
"atomic_index": {
  "id": "1170005876631875584",
  "name_en": "sum_total_amount",
  "name_ch": "总车费",
  "description": "",
  "create_by": null,
  "cal_exp": "sum(总车费) ",
  "cal_fn_ids": null,
  "l1_id": null,
  "l2_id": null,
  "l3_id": null,
  "table_id": "1169329693619625984",
  "tb_name": null,
  "dw_type": null,
  "field_ids": [ ],
  "field_names": null,
  "status": "DRAFT",
  "biz_type": "FACT_LOGIC_TABLE",
  "create_time": "2024-04-20T16:12:42.274+08:00",
  "update_time": "2024-04-20T16:12:42.274+08:00",
  "l1": null,
  "l2": null,
  "l3": null,
  "approval_info": null,
  "new_biz": null
},
"time_condition_name": null,
"create_time": "2023-12-04T09:37:49+08:00",
"update_time": "2023-12-04T09:37:49+08:00",
"l1": "test_001",
"l2": null,
"l3": null,
"summary_table_id": "1181167468975177728",
"approval_info": {
  "id": "1181167467817549824",
  "tenant_id": null,

```

```
"name_ch" : null,
"name_en" : null,
"biz_id" : "1181167466668310528",
"biz_type" : null,
"biz_info" : null,
"biz_info_obj" : null,
"biz_version" : 0,
"biz_status" : "PUBLISHED",
"approval_status" : "APPROVED",
"approval_type" : null,
"submit_time" : "2023-12-04T09:37:49+08:00",
"create_by" : null,
"l1" : null,
"l2" : null,
"l3" : null,
"approval_time" : "2023-12-04T09:37:49+08:00",
"approver" : "test_uesr",
"email" : null,
"msg" : "Fast Approval OK.",
"directory_path" : null
},
"new_biz" : null
}]
}
}
```

**状态码：400**

## BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

## Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

## Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDerivativeIndexesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDerivativeIndexesRequest request = new ListDerivativeIndexesRequest();
        try {
            ListDerivativeIndexesResponse response = client.listDerivativeIndexes(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = ListDerivativeIndexesRequest()
    response = client.list_derivative_indexes(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDerivativeIndexesRequest{}
    response, err := client.ListDerivativeIndexes(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是DerivativeIndexVO数组，以及总数。

状态码	描述
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.13.2 新建衍生指标

### 功能介绍

根据参数，新建衍生指标指标。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/derivative-indexs

表 8-688 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-689 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

表 8-690 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of <b>DerivativeIndexVO</b> objects	

表 8-691 DerivativeIndexVO

参数	是否必选	参数类型	描述
id	否	String	衍生指标ID, ID字符串。更新时必填。
name_en	是	String	衍生指标英文名称。
name_ch	是	String	衍生指标名称。
create_by	否	String	创建人。
data_type	否	String	字段类型。
l2_id	否	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	是	String	业务对象guid, ID字符串。
status	否	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>



参数	是否必选	参数类型	描述
atomic_index_id	是	String	原子指标ID, ID字符串。
time_condition_id	否	String	时间限定ID, ID字符串。
time_field_id	否	String	时间限定关联字段ID, ID字符串。
common_conditions	否	Array of <b>CommonConditionVO</b> objects	通用限定信息。
dimension_groups	否	Array of <b>DerivativeIndexDimensionVO</b> objects	统计维度, 更新时不可修改。
monitor	否	<b>MetricMonitorVO</b> object	指标监控。
l1	否	String	主题域分组中文名, 只读, 创建和更新时无需填写。
l2	否	String	主题域中文名, 只读, 创建和更新时无需填写。
l3	否	String	业务对象中文名, 只读, 创建和更新时无需填写。
approval_info	否	<b>ApprovalVO</b> object	审批信息, 只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	<b>BizVersionManageVO</b> object	业务版本管理, 只读。

表 8-692 CommonConditionVO

参数	是否必选	参数类型	描述
name	是	String	名称。
field_ids	是	Array of strings	字段id信息, 格式: table_id.field_id。
field_names	否	Array of strings	字段名称信息, 格式: 表名称. 字段名称。

参数	是否必选	参数类型	描述
cal_exp	是	String	计算表达式，形如'\$ {table_id.column_id} > 1'，其中table_id表示引用字段所属表ID，column_id表示引用字段ID。
cal_fn_ids	是	Array of strings	引用函数ID，ID字符串。
front_configs	否	String	前端表达式配置，用于前端数据恢复。

表 8-693 DerivativeIndexDimensionVO

参数	是否必选	参数类型	描述
group_id	是	String	维度分组ID。
role	否	String	维度角色。
dimension_id	否	String	维度ID，ID字符串。
hierarchies_id	否	String	维度层级ID，ID字符串。
group_name	否	String	维度分组名称。
group_code	否	String	维度分组编码。

参数	是否必选	参数类型	描述
biz_type	是	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。
dw_type	否	String	数据连接类型。

表 8-694 MetricMonitorVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
other_metric_ids	否	Array of strings	其他指标ID，ID字符串。

参数	是否必选	参数类型	描述
other_compo und_metric_id s	否	Array of strings	其他复合指标ID。
expression	否	String	告警表达式。
metric_id	否	String	挂载指ID, ID字符串。
front_configs	否	String	前端表达式配置, 用于前端数据 恢复。

参数	是否必选	参数类型	描述
metric_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

表 8-695 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>



参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-696 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

## 响应参数

状态码：200

表 8-697 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	创建衍生指标的返回结果，成功的个数。

表 8-698 data

参数	参数类型	描述
value	<b>BatchOperationVO</b> object	创建衍生指标操作返回结果。

表 8-699 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID，ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID，ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 运行中</li> <li>• SUCCESS: 操作成功</li> <li>• FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。
groups	Array of <b>BatchOperationVO</b> objects	分组信息。

**状态码：400**

表 8-700 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-701 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-702 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数进行衍生指标的创建。

POST https://{endpoint}/v2/{project\_id}/design/derivative-indexs

```
[ {
  "name_ch": "总车费(供应商,行程订单.车费)",
  "name_en": "sum_total_amount",
  "data_type": "STRING",
  "l3_id": "1169309252771299328",
  "atomic_index_id": "1170005876631875584",
  "dimension_groups": [ {
    "biz_type": "DIMENSION",
    "group_id": "1169322847009742848",
    "group_name": "供应商",
    "dimension_id": "1169322847009742848",
    "role": null,
    "hierarchies_id": null
  }, {
    "biz_type": "DEGENERATE_DIMENSION",
    "group_id": "1169329693619625984.1169329693695123463",
    "group_name": "行程订单.车费",
    "dimension_id": null,
    "role": null,
    "hierarchies_id": null
  } ],
  "time_condition_id": null,
  "time_field_id": null,
  "common_conditions": [ ],
  "monitor": null,
  "description": ""
} ]
```

## 响应示例

**状态码：200**

Success, 返回data数据是 BatchOperationVO。

```
{
  "data": {
```



```

"value": {
  "id": null,
  "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
  "group_id": "1300493983935508480",
  "biz_name": null,
  "biz_id": null,
  "operation_status": "SUCCESS",
  "operation_type": "BATCH_CREATE",
  "biz_info": null,
  "create_by": "test_uesr",
  "remark": null,
  "total": 1,
  "success": 1,
  "failed": 0,
  "rate": "1.00",
  "logs": null,
  "groups": [ {
    "id": "1300493984405270528",
    "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-9ce2cfd08fda4f3b95faa5ba65fa5ec3",
    "group_id": "1300493983935508480",
    "biz_name": "总车费(供应商,行程订单.车费)",
    "biz_id": "1300493984245886976",
    "operation_status": "SUCCESS",
    "operation_type": "BATCH_CREATE",
    "biz_info": "{\"atomic_index_id\":\"1170005876631875584\",\"create_by\":\"test_uesr\",\"data_type\":"
    "\":\"STRING\",\"description\":\"\",\"id\":\"1300493984245886976\",\"l3_id\":\"1169309252771299328\",\"name_ch\":"
    "\":\"总车费(供应商,行程订单.车费)\",\"name_en\":\"sum_total_amount\",\"status\":\"DRAFT\"}",
    "create_by": "test_uesr",
    "remark": "CREATE",
    "total": 0,
    "success": 0,
    "failed": 0,
    "rate": null,
    "logs": null,
    "groups": null
  } ]
}
}
}

```

### 状态码：400

BadRequest

```

{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}

```

### 状态码：401

Unauthorized

```

{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}

```

### 状态码：403

Forbidden

```

{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数进行衍生指标的创建。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateDesignDerivativeIndexSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateDesignDerivativeIndexRequest request = new CreateDesignDerivativeIndexRequest();
        List<DerivativeIndexDimensionVO> listBodyDimensionGroups = new ArrayList<>();
        listBodyDimensionGroups.add(
            new DerivativeIndexDimensionVO()
                .withGroupId("1169322847009742848")
                .withDimensionId("1169322847009742848")
                .withGroupName("供应商")
                .withBizType(DerivativeIndexDimensionVO.BizTypeEnum.fromValue("DIMENSION"))
        );
        listBodyDimensionGroups.add(
            new DerivativeIndexDimensionVO()
                .withGroupId("1169329693619625984.1169329693695123463")
                .withGroupName("行程订单.车费")
                .withBizType(DerivativeIndexDimensionVO.BizTypeEnum.fromValue("DEGENERATE_DIMENSION"))
        );
        List<DerivativeIndexVO> listbodyBody = new ArrayList<>();
        listbodyBody.add(
            new DerivativeIndexVO()
                .withNameEn("sum_total_amount")
                .withNameCh("总车费(供应商,行程订单.车费)")
                .withDescription("")
                .withDataType("STRING")
                .withL3Id("1169309252771299328")
                .withAtomicIndexId("1170005876631875584")
        );
    }
}
```

```
        .withCommonConditions()
        .withDimensionGroups(listBodyDimensionGroups)
    );
    request.withBody(listbodyBody);
    try {
        CreateDesignDerivativeIndexResponse response = client.createDesignDerivativeIndex(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

根据请求参数进行衍生指标的创建。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateDesignDerivativeIndexRequest()
        listDimensionGroupsBody = [
            DerivativeIndexDimensionVO(
                group_id="1169322847009742848",
                dimension_id="1169322847009742848",
                group_name="供应商",
                biz_type="DIMENSION"
            ),
            DerivativeIndexDimensionVO(
                group_id="1169329693619625984.1169329693695123463",
                group_name="行程订单.车费",
                biz_type="DEGENERATE_DIMENSION"
            )
        ]
        listBodybody = [
            DerivativeIndexVO(
                name_en="sum_total_amount",
                name_ch="总车费(供应商,行程订单.车费)",
            )
        ]
```

```
        description="",
        data_type="STRING",
        l3_id="1169309252771299328",
        atomic_index_id="1170005876631875584",
        dimension_groups=listDimensionGroupsBody
    )
]
request.body = listBodybody
response = client.create_design_derivative_index(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据请求参数进行衍生指标的创建。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDesignDerivativeIndexRequest{
        dimensionIdDimensionGroups:= "1169322847009742848"
        groupNameDimensionGroups:= "供应商"
        bizTypeDimensionGroups:= model.GetDerivativeIndexDimensionVoBizTypeEnum().DIMENSION
        groupNameDimensionGroups1:= "行程订单.车费"
        bizTypeDimensionGroups1:=
model.GetDerivativeIndexDimensionVoBizTypeEnum().DEGENERATE_DIMENSION
        var listDimensionGroupsBody = []model.DerivativeIndexDimensionVo{
            {
                GroupId: "1169322847009742848",
                DimensionId: &dimensionIdDimensionGroups,
                GroupName: &groupNameDimensionGroups,
                BizType: &bizTypeDimensionGroups,
            },
            {
                GroupId: "1169329693619625984.1169329693695123463",
                GroupName: &groupNameDimensionGroups1,
```

```

        BizType: &bizTypeDimensionGroups1,
    },
}
descriptionBody:= ""
dataTypeBody:= "STRING"
var listBodybody = []model.DerivativeIndexVo{
    {
        NameEn: "sum_total_amount",
        NameCh: "总车费(供应商,行程订单.车费)",
        Description: &descriptionBody,
        DataType: &dataTypeBody,
        L3Id: "1169309252771299328",
        AtomicIndexId: "1170005876631875584",
        DimensionGroups: &listDimensionGroupsBody,
    },
}
request.Body = &listBodybody
response, err := client.CreateDesignDerivativeIndex(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 BatchOperationVO。
400	BadRequest
401	Unauthorized
403	Forbidden

### 8.13.3 更新衍生指标

#### 功能介绍

根据传入参数，更新衍生指标。

#### 调用方法

请参见[如何调用API](#)。

#### URI

PUT /v2/{project\_id}/design/derivative-indexs

表 8-703 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-704 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-705 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	衍生指标ID，ID字符串。更新时必填。
name_en	是	String	衍生指标英文名称。
name_ch	是	String	衍生指标名称。
create_by	否	String	创建人。
data_type	否	String	字段类型。
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。

参数	是否必选	参数类型	描述
l3_id	是	String	业务对象guid, ID字符串。
status	否	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
atomic_index_id	是	String	原子指标ID, ID字符串。
time_condition_id	否	String	时间限定ID, ID字符串。
time_field_id	否	String	时间限定关联字段ID, ID字符串。
common_conditions	否	Array of <b>CommonConditionVO</b> objects	通用限定信息。
dimension_groups	否	Array of <b>DerivativeIndexDimensionVO</b> objects	统计维度, 更新时不可修改。
monitor	否	<b>MetricMonitorVO</b> object	指标监控。
l1	否	String	主题域分组中文名, 只读, 创建和更新时无需填写。
l2	否	String	主题域中文名, 只读, 创建和更新时无需填写。
l3	否	String	业务对象中文名, 只读, 创建和更新时无需填写。
approval_info	否	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。

参数	是否必选	参数类型	描述
new_biz	否	BizVersionManagementVO object	业务版本管理，只读。

表 8-706 CommonConditionVO

参数	是否必选	参数类型	描述
name	是	String	名称。
field_ids	是	Array of strings	字段id信息，格式：table_id.field_id。
field_names	否	Array of strings	字段名称信息，格式：表名称.字段名称。
cal_exp	是	String	计算表达式，形如'\$ {table_id.column_id} > 1'，其中table_id表示引用字段所属表ID，column_id表示引用字段ID。
cal_fn_ids	是	Array of strings	引用函数ID，ID字符串。
front_configs	否	String	前端表达式配置，用于前端数据恢复。

表 8-707 DerivativeIndexDimensionVO

参数	是否必选	参数类型	描述
group_id	是	String	维度分组ID。
role	否	String	维度角色。
dimension_id	否	String	维度ID，ID字符串。
hierarchies_id	否	String	维度层级ID，ID字符串。
group_name	否	String	维度分组名称。
group_code	否	String	维度分组编码。



参数	是否必选	参数类型	描述
biz_type	是	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。
dw_type	否	String	数据连接类型。

表 8-708 MetricMonitorVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
other_metric_ids	否	Array of strings	其他指标ID，ID字符串。

参数	是否必选	参数类型	描述
other_compo und_metric_id s	否	Array of strings	其他复合指标ID。
expression	否	String	告警表达式。
metric_id	否	String	挂载指ID, ID字符串。
front_configs	否	String	前端表达式配置, 用于前端数据 恢复。

参数	是否必选	参数类型	描述
metric_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

表 8-709 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。



表 8-710 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

## 响应参数

状态码：200

表 8-711 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	更新衍生指标的返回结果，成功的个数。

表 8-712 data

参数	参数类型	描述
value	<b>DerivativeIndexVO</b> object	更新衍生指标操作返回结果。

表 8-713 DerivativeIndexVO

参数	参数类型	描述
id	String	衍生指标ID，ID字符串。更新时必填。
name_en	String	衍生指标英文名称。
name_ch	String	衍生指标名称。
description	String	描述，只读。
create_by	String	创建人。

参数	参数类型	描述
data_type	String	字段类型。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象guid, ID字符串。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
atomic_index_id	String	原子指标ID, ID字符串。
time_condition_id	String	时间限定ID, ID字符串。
time_field_id	String	时间限定关联字段ID, ID字符串。
time_field_name	String	时间限定关联字段名称, 只读。
common_conditions	Array of <b>CommonConditionVO</b> objects	通用限定信息。
dimension_groups	Array of <b>DerivativeIndexDimensionVO</b> objects	统计维度, 更新时不可修改。
monitor	<b>MetricMonitorVO</b> object	指标监控。
atomic_index	<b>AtomicIndexVO</b> object	原子指标信息, 名称、单位, 只读。
time_condition_name	String	时间限定名称, 只读。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
l1	String	主题域分组中文名, 只读, 创建和更新时无需填写。
l2	String	主题域中文名, 只读, 创建和更新时无需填写。
l3	String	业务对象中文名, 只读, 创建和更新时无需填写。
summary_table_id	String	汇总表ID, 只读, ID字符串。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理, 只读。

表 8-714 CommonConditionVO

参数	参数类型	描述
name	String	名称。
field_ids	Array of strings	字段id信息, 格式: table_id.field_id。
field_names	Array of strings	字段名称信息, 格式: 表名称.字段名称。
cal_exp	String	计算表达式, 形如'\$ {table_id.column_id} > 1', 其中table_id表示引用字段所属表ID, column_id表示引用字段ID。
cal_fn_ids	Array of strings	引用函数ID, ID字符串。
front_configs	String	前端表达式配置, 用于前端数据恢复。
id	String	通用限定ID, 只读, ID字符串。

表 8-715 DerivativeIndexDimensionVO

参数	参数类型	描述
group_id	String	维度分组ID。

参数	参数类型	描述
role	String	维度角色。
dimension_id	String	维度ID, ID字符串。
hierarchies_id	String	维度层级ID, ID字符串。
ordinal	Integer	序号, 只读。
group_name	String	维度分组名称。
group_code	String	维度分组编码。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
hierarchies	Array of <a href="#">DimensionHierarchiesVO</a> objects	层级属性，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，只读，ID字符串。
dw_type	String	数据连接类型。
id	String	层级的ID，只读，ID字符串。

表 8-716 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	层级名称。

参数	参数类型	描述
attrs	Array of <b>HierarchiesAttrVO</b> objects	层级包含的属性。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-717 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码，ID字符串。
hierarchies_id	String	层级ID，ID字符串。
attr_id	String	属性ID，ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称，只读。
detail_attr_ids	Array of strings	详情属性ID，ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文，只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段，只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段，只读。

表 8-718 DimensionAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
dimension_id	String	维度ID，只读，ID字符串。
code_table_field_id	String	码表属性ID，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。

表 8-719 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-720 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理, 只读。

表 8-721 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-722 MetricMonitorVO

参数	参数类型	描述
id	String	编码, ID字符串。
other_metric_ids	Array of strings	其他指标ID, ID字符串。
other_metric_names	Array of strings	其他指标名称, 只读。

参数	参数类型	描述
other_compound_metric_ids	Array of strings	其他复合指标ID。
other_compound_metric_names	Array of strings	其他复合指标名称。
expression	String	告警表达式。
metric_id	String	挂载指ID, ID字符串。
front_configs	String	前端表达式配置, 用于前端数据恢复。



参数	参数类型	描述
metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

表 8-723 AtomicIndexVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	原子指标英文名。
name_ch	String	原子指标中文名。
description	String	描述。
create_by	String	创建人。
cal_exp	String	计算表达式，形如'sum(\${fact_column_id})'，其中fact_column_id表示引用事实表中的字段ID。
cal_fn_ids	Array of strings	引用函数ID，ID字符串。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象guid，ID字符串。
table_id	String	事实表ID，ID字符串。
tb_name	String	事实表名称。

参数	参数类型	描述
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
field_ids	Array of strings	字段ID信息，ID字符串。
field_names	Array of strings	字段名称信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。

表 8-724 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。



参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-725 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-726 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-727 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-728 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入的参数，更新衍生指标。

PUT https://{endpoint}/v2/{project\_id}/design/derivative-idxs

```
{
  "id": "1300493984245886976",
  "name_en": "sum_total_amount",
  "name_ch": "总车费(供应商,行程订单.车费)",
  "data_type": "STRING",
  "l3_id": "1169309252771299328",
  "atomic_index_id": "1170005876631875584",
  "time_condition_id": null,
  "time_field_id": null,
  "common_conditions": [],
  "dimension_groups": [ {
    "ordinal": 1,
    "biz_type": "DIMENSION",
    "group_id": "1169322847009742848",
```

```

"dimension_id" : "1169322847009742848",
"role" : null,
"hierarchies_id" : null
}, {
"ordinal" : 2,
"biz_type" : "DEGENERATE_DIMENSION",
"group_id" : "1169329693619625984.1169329693695123463",
"dimension_id" : null,
"role" : null,
"hierarchies_id" : null
}],
"monitor" : null,
"description" : ""
}

```

## 响应示例

**状态码：200**

Success, 返回data数据是 DerivativeIndexVO。

```

{
"value" : {
"id" : "1300493984245886976",
"name_en" : "sum_total_amount",
"name_ch" : "总车费(供应商,行程订单.车费)",
"description" : "",
"create_by" : "test_user",
"data_type" : "STRING",
"l1_id" : null,
"l2_id" : null,
"l3_id" : "1169309252771299328",
"status" : "DRAFT",
"atomic_index_id" : "1170005876631875584",
"time_condition_id" : null,
"time_field_id" : null,
"time_field_name" : null,
"common_conditions" : [ ],
"dimension_groups" : [ {
"group_id" : "1169322847009742848",
"role" : null,
"dimension_id" : "1169322847009742848",
"hierarchies_id" : null,
"ordinal" : 1,
"group_name" : null,
"group_code" : null,
"biz_type" : "DIMENSION",
"hierarchies" : null,
"l1" : null,
"l2" : null,
"l3" : null,
"l1_id" : null,
"l2_id" : null,
"l3_id" : null,
"dw_type" : null,
"id" : null,
"from_public" : null
}, {
"group_id" : "1169329693619625984.1169329693695123463",
"role" : null,
"dimension_id" : null,
"hierarchies_id" : null,
"ordinal" : 2,
"group_name" : null,
"group_code" : null,
"biz_type" : "DEGENERATE_DIMENSION",
"hierarchies" : null,
"l1" : null,
"l2" : null,

```

```

    "l3" : null,
    "l1_id" : null,
    "l2_id" : null,
    "l3_id" : null,
    "dw_type" : null,
    "id" : null,
    "from_public" : null
  } ],
  "monitor" : null,
  "atomic_index" : null,
  "time_condition_name" : null,
  "create_time" : "2024-10-28T16:18:48+08:00",
  "update_time" : "2024-10-28T17:40:14.903+08:00",
  "l1" : null,
  "l2" : null,
  "l3" : null,
  "summary_table_id" : null,
  "approval_info" : null,
  "new_biz" : null
}
}

```

### 状态码：400

BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

### 状态码：401

Unauthorized

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}

```

### 状态码：403

Forbidden

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

根据传入的参数，更新衍生指标。

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

```

```
import java.util.List;
import java.util.ArrayList;

public class UpdateDesignDerivativeIndexSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateDesignDerivativeIndexRequest request = new UpdateDesignDerivativeIndexRequest();
        DerivativeIndexVO body = new DerivativeIndexVO();
        List<DerivativeIndexDimensionVO> listbodyDimensionGroups = new ArrayList<>();
        listbodyDimensionGroups.add(
            new DerivativeIndexDimensionVO()
                .withGroupId("1169322847009742848")
                .withDimensionId("1169322847009742848")
                .withOrdinal(1)
                .withBizType(DerivativeIndexDimensionVO.BizTypeEnum.fromValue("DIMENSION"))
        );
        listbodyDimensionGroups.add(
            new DerivativeIndexDimensionVO()
                .withGroupId("1169329693619625984.1169329693695123463")
                .withOrdinal(2)
                .withBizType(DerivativeIndexDimensionVO.BizTypeEnum.fromValue("DEGENERATE_DIMENSION"))
        );
        body.withDimensionGroups(listbodyDimensionGroups);
        body.withAtomicIndexId("1170005876631875584");
        body.withL3Id("1169309252771299328");
        body.withDataType("STRING");
        body.withDescription("");
        body.withNameCh("总车费(供应商,行程订单,车费)");
        body.withNameEn("sum_total_amount");
        body.withId("1300493984245886976");
        request.withBody(body);
        try {
            UpdateDesignDerivativeIndexResponse response = client.updateDesignDerivativeIndex(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据传入的参数，更新衍生指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateDesignDerivativeIndexRequest()
        listDimensionGroupsbody = [
            DerivativeIndexDimensionVO(
                group_id="1169322847009742848",
                dimension_id="1169322847009742848",
                ordinal=1,
                biz_type="DIMENSION"
            ),
            DerivativeIndexDimensionVO(
                group_id="1169329693619625984.1169329693695123463",
                ordinal=2,
                biz_type="DEGENERATE_DIMENSION"
            )
        ]
        request.body = DerivativeIndexVO(
            dimension_groups=listDimensionGroupsbody,
            atomic_index_id="1170005876631875584",
            l3_id="1169309252771299328",
            data_type="STRING",
            description="",
            name_ch="总车费(供应商,行程订单.车费)",
            name_en="sum_total_amount",
            id="1300493984245886976"
        )
        response = client.update_design_derivative_index(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据传入的参数，更新衍生指标。

```
package main
```



```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDesignDerivativeIndexRequest{}
    dimensionIdDimensionGroups:= "1169322847009742848"
    ordinalDimensionGroups:= int32(1)
    bizTypeDimensionGroups:= model.GetDerivativeIndexDimensionVoBizTypeEnum().DIMENSION
    ordinalDimensionGroups1:= int32(2)
    bizTypeDimensionGroups1:=
model.GetDerivativeIndexDimensionVoBizTypeEnum().DEGENERATE_DIMENSION
var listDimensionGroupsbody = []model.DerivativeIndexDimensionVo{
    {
        GroupId: "1169322847009742848",
        DimensionId: &dimensionIdDimensionGroups,
        Ordinal: &ordinalDimensionGroups,
        BizType: &bizTypeDimensionGroups,
    },
    {
        GroupId: "1169329693619625984.1169329693695123463",
        Ordinal: &ordinalDimensionGroups1,
        BizType: &bizTypeDimensionGroups1,
    },
}
    dataTypeDerivativeIndexVo:= "STRING"
    descriptionDerivativeIndexVo:= ""
    idDerivativeIndexVo:= "1300493984245886976"
    request.Body = &model.DerivativeIndexVo{
        DimensionGroups: &listDimensionGroupsbody,
        AtomicIndexId: "1170005876631875584",
        L3Id: "1169309252771299328",
        DataType: &dataTypeDerivativeIndexVo,
        Description: &descriptionDerivativeIndexVo,
        NameCh: "总车费(供应商,行程订单.车费)",
        NameEn: "sum_total_amount",
        Id: &idDerivativeIndexVo,
    }
    response, err := client.UpdateDesignDerivativeIndex(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 DerivativeIndexVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.13.4 删除衍生指标

### 功能介绍

根据衍生指标ID，删除衍生指标。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/derivative-indexs

表 8-729 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-730 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-731 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-732 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-733 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-734 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-735 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-736 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入的参数，删除衍生指标，只支持删除状态为草稿、已下线、已驳回的衍生指标。

```
DELETE https://{endpoint}/v2/{project_id}/design/derivative-indexs
{
  "ids" : [ "1227990857618227200" ]
}
```

## 响应示例

### 状态码：200

Success，返回data数据是删除个数。

```
{
  "value" : 1
}
```

### 状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据传入的参数，删除衍生指标，只支持删除状态为草稿、已下线、已驳回的衍生指标。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteDesignDerivativeIndexSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteDesignDerivativeIndexRequest request = new DeleteDesignDerivativeIndexRequest();
        IdsParam body = new IdsParam();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("1227990857618227200");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            DeleteDesignDerivativeIndexResponse response = client.deleteDesignDerivativeIndex(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据传入的参数，删除衍生指标，只支持删除状态为草稿、已下线、已驳回的衍生指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteDesignDerivativeIndexRequest()
    listIdsbody = [
        "1227990857618227200"
    ]
    request.body = IdsParam(
        ids=listIdsbody
    )
    response = client.delete_design_derivative_index(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据传入的参数，删除衍生指标，只支持删除状态为草稿、已下线、已驳回的衍生指标。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDesignDerivativeIndexRequest{}
```

```
var listIdsbody = []string{
    "1227990857618227200",
}
request.Body = &model.IdsParam{
    Ids: listIdsbody,
}
response, err := client.DeleteDesignDerivativeIndex(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.13.5 查看衍生指标详情

### 功能介绍

通过ID获取衍生详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/derivative-indexs/{id}

表 8-737 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。



表 8-738 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

## 请求参数

表 8-739 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-740 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	返回数据。

表 8-741 data

参数	参数类型	描述
value	<a href="#">DerivativeIndexVO</a> object	返回的数据信息。

表 8-742 DerivativeIndexVO

参数	参数类型	描述
id	String	衍生指标ID, ID字符串。更新时必填。
name_en	String	衍生指标英文名称。
name_ch	String	衍生指标名称。
description	String	描述, 只读。
create_by	String	创建人。
data_type	String	字段类型。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象guid, ID字符串。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
atomic_index_id	String	原子指标ID, ID字符串。
time_condition_id	String	时间限定ID, ID字符串。
time_field_id	String	时间限定关联字段ID, ID字符串。
time_field_name	String	时间限定关联字段名称, 只读。
common_conditions	Array of <b>CommonConditionVO</b> objects	通用限定信息。
dimension_groups	Array of <b>DerivativeIndexDimensionVO</b> objects	统计维度, 更新时不可修改。
monitor	<b>MetricMonitorVO</b> object	指标监控。

参数	参数类型	描述
atomic_index	<b>AtomicIndexVO</b> object	原子指标信息，名称、单位，只读。
time_condition_name	String	时间限定名称，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
summary_table_id	String	汇总表ID，只读，ID字符串。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-743 CommonConditionVO

参数	参数类型	描述
name	String	名称。
field_ids	Array of strings	字段id信息，格式：table_id.field_id。
field_names	Array of strings	字段名称信息，格式：表名称.字段名称。
cal_exp	String	计算表达式，形如'\$ {table_id.column_id} > 1'，其中table_id表示引用字段所属表ID，column_id表示引用字段ID。
cal_fn_ids	Array of strings	引用函数ID，ID字符串。
front_configs	String	前端表达式配置，用于前端数据恢复。

参数	参数类型	描述
id	String	通用限定ID，只读，ID字符串。

表 8-744 DerivativeIndexDimensionVO

参数	参数类型	描述
group_id	String	维度分组ID。
role	String	维度角色。
dimension_id	String	维度ID，ID字符串。
hierarchies_id	String	维度层级ID，ID字符串。
ordinal	Integer	序号，只读。
group_name	String	维度分组名称。
group_code	String	维度分组编码。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
hierarchies	Array of <a href="#">DimensionHierarchiesVO</a> objects	层级属性，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，只读，ID字符串。
dw_type	String	数据连接类型。
id	String	层级的ID，只读，ID字符串。

表 8-745 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	层级名称。

参数	参数类型	描述
attrs	Array of <b>HierarchiesAttrVO</b> objects	层级包含的属性。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-746 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码，ID字符串。
hierarchies_id	String	层级ID，ID字符串。
attr_id	String	属性ID，ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称，只读。
detail_attr_ids	Array of strings	详情属性ID，ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文，只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段，只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段，只读。

表 8-747 DimensionAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
dimension_id	String	维度ID，只读，ID字符串。
code_table_field_id	String	码表属性ID，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级



参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。

表 8-748 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-749 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理, 只读。

表 8-750 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-751 MetricMonitorVO

参数	参数类型	描述
id	String	编码, ID字符串。
other_metric_ids	Array of strings	其他指标ID, ID字符串。
other_metric_names	Array of strings	其他指标名称, 只读。

参数	参数类型	描述
other_compound_metric_ids	Array of strings	其他复合指标ID。
other_compound_metric_names	Array of strings	其他复合指标名称。
expression	String	告警表达式。
metric_id	String	挂载指ID, ID字符串。
front_configs	String	前端表达式配置, 用于前端数据恢复。

参数	参数类型	描述
metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

表 8-752 AtomicIndexVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	原子指标英文名。
name_ch	String	原子指标中文名。
description	String	描述。
create_by	String	创建人。
cal_exp	String	计算表达式，形如'sum(\${fact_column_id})'，其中fact_column_id表示引用事实表中的字段ID。
cal_fn_ids	Array of strings	引用函数ID，ID字符串。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象guid，ID字符串。
table_id	String	事实表ID，ID字符串。
tb_name	String	事实表名称。

参数	参数类型	描述
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
field_ids	Array of strings	字段ID信息，ID字符串。
field_names	Array of strings	字段名称信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。

表 8-753 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-754 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400



表 8-755 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-756 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-757 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据衍生指标ID，获取衍生指标详情。

GET https://{endpoint}/v2/{project\_id}/design/derivative-indexs/1181167466668310528

## 响应示例

状态码：200

Success，返回data数据是DerivativeIndexVO详情。

```
{
  "data" : {
    "value" : {
      "id" : "1181167466668310528",
      "name_en" : "sum_total_amount",
      "name_ch" : "总车费(供应商)",
      "description" : ""
    }
  }
}
```

```

"create_by" : "test_uesr",
"data_type" : "STRING",
"l1_id" : "1170066794946420736",
"l2_id" : null,
"l3_id" : null,
"status" : "PUBLISHED",
"atomic_index_id" : "1170005876631875584",
"time_condition_id" : null,
"time_field_id" : null,
"time_field_name" : null,
"common_conditions" : [ ],
"dimension_groups" : [ {
  "group_id" : "1169322847009742848",
  "role" : null,
  "dimension_id" : "1169322847009742848",
  "hierarchies_id" : null,
  "ordinal" : 1,
  "group_name" : "供应商",
  "group_code" : "dim_vendor",
  "biz_type" : "DIMENSION",
  "hierarchies" : null,
  "l1" : null,
  "l2" : null,
  "l3" : null,
  "l1_id" : null,
  "l2_id" : null,
  "l3_id" : null,
  "dw_type" : null,
  "id" : "1181167466768973824"
}],
"monitor" : null,
"atomic_index" : {
  "id" : "1170005876631875584",
  "name_en" : "sum_total_amount",
  "name_ch" : "总车费",
  "description" : "",
  "create_by" : null,
  "cal_exp" : "sum(总车费)",
  "cal_fn_ids" : null,
  "l1_id" : null,
  "l2_id" : null,
  "l3_id" : null,
  "table_id" : "1169329693619625984",
  "tb_name" : null,
  "dw_type" : null,
  "field_ids" : [ ],
  "field_names" : null,
  "status" : "DRAFT",
  "biz_type" : "FACT_LOGIC_TABLE",
  "create_time" : "2024-04-20T16:15:41.487+08:00",
  "update_time" : "2024-04-20T16:15:41.487+08:00",
  "l1" : null,
  "l2" : null,
  "l3" : null,
  "approval_info" : null,
  "new_biz" : null
},
"time_condition_name" : null,
"create_time" : "2023-12-04T09:37:49+08:00",
"update_time" : "2023-12-04T09:37:49+08:00",
"l1" : "test_001",
"l2" : null,
"l3" : null,
"summary_table_id" : "1181167468975177728",
"approval_info" : {
  "id" : "1181167467817549824",
  "tenant_id" : null,
  "name_ch" : null,
  "name_en" : null,

```

```

    "biz_id": "1181167466668310528",
    "biz_type": null,
    "biz_info": null,
    "biz_info_obj": null,
    "biz_version": 0,
    "biz_status": "PUBLISHED",
    "approval_status": "APPROVED",
    "approval_type": null,
    "submit_time": "2023-12-04T09:37:49+08:00",
    "create_by": null,
    "l1": null,
    "l2": null,
    "l3": null,
    "approval_time": "2023-12-04T09:37:49+08:00",
    "approver": "test_uesr",
    "email": null,
    "msg": "Fast Approval OK.",
    "directory_path": null
  },
  "new_biz": null
}
}
}

```

#### 状态码：400

BadRequest

```

{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}

```

#### 状态码：401

Unauthorized

```

{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}

```

#### 状态码：403

Forbidden

```

{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

```

```
public class ShowDerivativeIndexByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDerivativeIndexByIdRequest request = new ShowDerivativeIndexByIdRequest();
        request.withId("{id}");
        try {
            ShowDerivativeIndexByIdResponse response = client.showDerivativeIndexById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```

request = ShowDerivativeIndexByIdRequest()
request.id = "{id}"
response = client.show_derivative_index_by_id(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDerivativeIndexByIdRequest{}
    request.Id = "{id}"
    response, err := client.ShowDerivativeIndexById(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是DerivativeIndexVO详情。

状态码	描述
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.14 复合指标接口

### 8.14.1 查找复合指标

#### 功能介绍

通过中英文名称、创建者、审核人、状态、修改时间、l3Id分页查找复合指标信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/compound-metrics

表 8-758 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-759 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。

参数	是否必选	参数类型	描述
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
dimension_group	否	String	依据维度颗粒度查维度属性。
atomic_index_id	否	String	依据原子指标ID查维度属性，ID字符串。
l3_id	否	String	业务对象l3的ID，ID字符串。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-760 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-761 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	返回数据。

表 8-762 data

参数	参数类型	描述
value	<a href="#">value</a> object	返回的数据信息。



表 8-763 value

参数	参数类型	描述
records	Array of <b>CompoundMetricVO</b> objects	CompoundMetricVO数组。
total	Integer	总数。

表 8-764 CompoundMetricVO

参数	参数类型	描述
id	String	复合指标ID，ID字符串。更新时必须填。
name_en	String	复合指标英文名称。
name_ch	String	复合指标名称。
description	String	描述
dimension_group	String	统计维度ID，更新时不可修改此值。
group_name	String	颗粒度名称，只读。
group_code	String	颗粒度编码，只读。
compound_type	String	复合指标类型。 枚举值： <ul style="list-style-type: none"> <li>EXPRESSION: 表达式</li> <li>PERIODICITY_VALUED_COMPARISON: 环比增长率</li> <li>INTERVAL_VALUED_COMPARISON: 同比增长率</li> </ul>
comparison_type	String	同比配置，复合指标类型为“同比增长率”时必须填。 枚举值： <ul style="list-style-type: none"> <li>YEAR_TO_YEAR: 年同比</li> <li>MONTH_TO_MONTH: 月同比</li> <li>WEEK_TO_WEEK: 周同比</li> </ul>
metric_ids	Array of strings	设定衍生指标，复合指标类型为“同比增长率”或“环比增长率”时必须填，ID字符串。
metric_names	Array of strings	指标名称信息。
compound_metric_ids	Array of strings	复合指标信息，ID字符串。

参数	参数类型	描述
compound_metric_names	Array of strings	复合指标名称信息。
cal_fn_ids	Array of strings	引用函数ID, ID字符串。
cal_exp	String	计算表达式, 复合指标类型为“表达式”时必填。形如\${index_id} + \${compound#index_id}, 其中index_id代表引用的衍生指标ID, compound#index_id代表引用的复合指标ID。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象ID, ID字符串。
data_type	String	字段类型。
create_by	String	创建人。
update_by	String	更新人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理, 只读。

参数	参数类型	描述
monitor	<b>MetricMonitorVO</b> object	指标监控信息。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
summary_table_id	String	汇总表ID，只读，ID字符串。

表 8-765 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-766 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-767 MetricMonitorVO

参数	参数类型	描述
id	String	编码，ID字符串。
other_metric_ids	Array of strings	其他指标ID，ID字符串。
other_metric_names	Array of strings	其他指标名称，只读。
other_compound_metric_ids	Array of strings	其他复合指标ID。
other_compound_metric_names	Array of strings	其他复合指标名称。
expression	String	告警表达式。
metric_id	String	挂载指ID，ID字符串。
front_configs	String	前端表达式配置，用于前端数据恢复。

参数	参数类型	描述
metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

**状态码：400**

表 8-768 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-769 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-770 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据查询条件，分页查询复合指标列表。

GET https://{endpoint}/v2/{project\_id}/design/compound-metrics?offset=1&limit=10&name=com\_

## 响应示例

**状态码：200**

Success，返回data数据是CompoundMetricVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1170056628536553472",
        "name_en": "com_compound_metric_test",
        "name_ch": "复合指标测试",
        "description": "",
        "dimension_group": "1169322847009742848,1169329693619625984.1169329693695123460.1169327565677182976",
        "group_name": "供应商,行程订单.下车时间",
        "group_code": null,
        "compound_type": "EXPRESSION",
        "comparison_type": null,
        "metric_ids": [ "1170007051066355712" ],
        "metric_names": null,
        "compound_metric_ids": [ ],
        "compound_metric_names": null,
        "cal_fn_ids": null,
        "cal_exp": "${1170007051066355712}/2",
        "l1_id": "1169309252771299328",
        "l2_id": "1169309374670356480,1169310587398889472",
        "l3_id": null,
        "data_type": "BIGINT",
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "status": "PUBLISHED",
        "create_time": "2023-11-03T17:47:19+08:00",
        "update_time": "2023-11-03T17:55:12+08:00",
        "approval_info": {
          "id": "1170056629291528192",
          "tenant_id": null,
          "name_ch": null,
          "name_en": null,
          "biz_id": "1170056628536553472",
          "biz_type": null,
          "biz_info": null,
          "biz_info_obj": null,
          "biz_version": 0,
          "biz_status": "PUBLISHED",
          "approval_status": "APPROVED",
```

```
"approval_type" : null,
"submit_time" : "2023-11-03T17:47:19+08:00",
"create_by" : null,
"l1" : null,
"l2" : null,
"l3" : null,
"approval_time" : "2023-11-03T17:55:12+08:00",
"approver" : "test_uesr",
"email" : null,
"msg" : "Fast Approval OK.",
"directory_path" : null
},
"new_biz" : null,
"monitor" : null,
"l1" : "城市交通",
"l2" : "行程记录/行程事实",
"l3" : null,
"summary_table_id" : "1170007052639219712"
}]
}
}
```

#### 状态码：400

##### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

##### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

##### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class ListCompoundMetricsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListCompoundMetricsRequest request = new ListCompoundMetricsRequest();  
        try {  
            ListCompoundMetricsResponse response = client.listCompoundMetrics(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.getenv("CLOUD_SDK_AK")  
    sk = os.getenv("CLOUD_SDK_SK")  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListCompoundMetricsRequest()  
        response = client.list_compound_metrics(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCompoundMetricsRequest{}
    response, err := client.ListCompoundMetrics(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是CompoundMetricVO数组，以及总数。
400	BadRequest
401	Unauthorized

状态码	描述
403	Forbidden

## 8.14.2 新建复合指标

### 功能介绍

根据参数，新建复合指标。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/compound-metrics

表 8-771 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-772 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。



参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

表 8-773 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	复合指标ID, ID字符串。更新时必须填。
name_en	是	String	复合指标英文名称。
name_ch	是	String	复合指标名称。
description	否	String	描述
dimension_group	是	String	统计维度ID, 更新时不可修改此值。
compound_type	是	String	复合指标类型。 枚举值: <ul style="list-style-type: none"> <li>• EXPRESSION: 表达式</li> <li>• PERIODICITY_VALUED_COMPARISON: 环比增长率</li> <li>• INTERVAL_VALUED_COMPARISON: 同比增长率</li> </ul>
comparison_type	否	String	同比配置, 复合指标类型为“同比增长率”时必须填。 枚举值: <ul style="list-style-type: none"> <li>• YEAR_TO_YEAR: 年同比</li> <li>• MONTH_TO_MONTH: 月同比</li> <li>• WEEK_TO_WEEK: 周同比</li> </ul>
metric_ids	否	Array of strings	设定衍生指标, 复合指标类型为“同比增长率”或“环比增长率”时必须填, ID字符串。
metric_names	否	Array of strings	指标名称信息。
compound_metric_ids	否	Array of strings	复合指标信息, ID字符串。

参数	是否必选	参数类型	描述
compound_metric_names	否	Array of strings	复合指标名称信息。
cal_fn_ids	否	Array of strings	引用函数ID, ID字符串。
cal_exp	否	String	计算表达式, 复合指标类型为“表达式”时必填。形如\${index_id} + \${compound#index_id}, 其中index_id代表引用的衍生指标ID, compound#index_id代表引用的复合指标ID。
l2_id	否	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	是	String	业务对象ID, ID字符串。
data_type	是	String	字段类型。
create_by	否	String	创建人。
update_by	否	String	更新人。
status	否	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManageVO object	业务版本管理, 只读。
monitor	否	MetricMonitorVO object	指标监控信息。
l1	否	String	主题域分组中文名, 只读, 创建和更新时无需填写。

参数	是否必选	参数类型	描述
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。

表 8-774 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-775 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>



参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-776 MetricMonitorVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
other_metric_ids	否	Array of strings	其他指标ID，ID字符串。
other_compond_metric_ids	否	Array of strings	其他复合指标ID。
expression	否	String	告警表达式。
metric_id	否	String	挂载指ID，ID字符串。
front_configs	否	String	前端表达式配置，用于前端数据恢复。

参数	是否必选	参数类型	描述
metric_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

## 响应参数

状态码：200

表 8-777 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	创建复合指标的返回结果，成功的个数。

表 8-778 data

参数	参数类型	描述
value	<b>CompoundMetricVO</b> object	创建复合指标操作返回结果。

表 8-779 CompoundMetricVO

参数	参数类型	描述
id	String	复合指标ID, ID字符串。更新时必填。
name_en	String	复合指标英文名称。
name_ch	String	复合指标名称。
description	String	描述
dimension_group	String	统计维度ID, 更新时不可修改此值。
group_name	String	颗粒度名称, 只读。
group_code	String	颗粒度编码, 只读。
compound_type	String	复合指标类型。 枚举值: <ul style="list-style-type: none"> <li>• EXPRESSION: 表达式</li> <li>• PERIODICITY_VALUED_COMPARISON: 环比增长率</li> <li>• INTERVAL_VALUED_COMPARISON: 同比增长率</li> </ul>
comparison_type	String	同比配置, 复合指标类型为“同比增长率”时必填。 枚举值: <ul style="list-style-type: none"> <li>• YEAR_TO_YEAR: 年同比</li> <li>• MONTH_TO_MONTH: 月同比</li> <li>• WEEK_TO_WEEK: 周同比</li> </ul>
metric_ids	Array of strings	设定衍生指标, 复合指标类型为“同比增长率”或“环比增长率”时必填, ID字符串。
metric_names	Array of strings	指标名称信息。
compound_metric_ids	Array of strings	复合指标信息, ID字符串。
compound_metric_names	Array of strings	复合指标名称信息。
cal_fn_ids	Array of strings	引用函数ID, ID字符串。

参数	参数类型	描述
cal_exp	String	计算表达式，复合指标类型为“表达式”时必填。形如\${index_id} + \${compound#index_id}，其中index_id代表引用的衍生指标ID，compound#index_id代表引用的复合指标ID。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
data_type	String	字段类型。
create_by	String	创建人。
update_by	String	更新人。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
monitor	<b>MetricMonitorVO</b> object	指标监控信息。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
summary_table_id	String	汇总表ID，只读，ID字符串。

表 8-780 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-781 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-782 MetricMonitorVO

参数	参数类型	描述
id	String	编码，ID字符串。
other_metric_ids	Array of strings	其他指标ID，ID字符串。
other_metric_names	Array of strings	其他指标名称，只读。
other_compound_metric_ids	Array of strings	其他复合指标ID。
other_compound_metric_names	Array of strings	其他复合指标名称。
expression	String	告警表达式。
metric_id	String	挂载指ID，ID字符串。
front_configs	String	前端表达式配置，用于前端数据恢复。

参数	参数类型	描述
metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

**状态码：400**

表 8-783 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-784 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-785 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据下列参数，新建复合指标。

POST https://{endpoint}/v2/{project\_id}/design/compound-metrics

```
{
  "id" : 0,
  "name_ch" : "测试复合指标",
  "name_en" : "test_compound_index",
  "l3_id" : "1169309252771299328",
  "dimension_group" : "1169322847009742848",
  "data_type" : "STRING",
  "compound_type" : "EXPRESSION",
  "monitor" : {
    "id" : null,
    "other_metric_ids" : [ ],
    "other_compound_metric_ids" : [ ],
    "expression" : "",
    "metric_id" : null,
    "metric_type" : "COMPOUND_METRIC"
  },
  "description" : "",
  "metric_ids" : [ "1300493917942329344", "1268168163686780928", "1181167466668310528" ],
  "compound_metric_ids" : [ ],
  "cal_exp" : "${1300493917942329344}+${1268168163686780928}+${1181167466668310528}"
}
```

## 响应示例

**状态码：200**

Success，返回data数据是 CompoundMetricVO。

```
{
  "value" : {
    "id" : "1300537267533774848",
    "name_en" : "test_compound_index",
    "name_ch" : "测试复合指标",
    "description" : "",
    "dimension_group" : "1169322847009742848",
    "group_name" : null,
    "group_code" : null,
    "compound_type" : "EXPRESSION",
    "comparison_type" : null,
    "metric_ids" : [ "1300493917942329344", "1268168163686780928", "1181167466668310528" ],
    "metric_names" : null,
    "compound_metric_ids" : [ ],
    "compound_metric_names" : null,
    "cal_fn_ids" : null,
    "cal_exp" : "${1300493917942329344}+${1268168163686780928}+${1181167466668310528}",
    "l1_id" : null,
    "l2_id" : null,
    "l3_id" : "1169309252771299328",
  }
}
```



```
"data_type": "STRING",
"create_by": "test_user",
"update_by": "test_user",
"status": "DRAFT",
"create_time": "2024-10-28T19:10:47+08:00",
"update_time": "2024-10-28T19:20:23.085+08:00",
"approval_info": null,
"new_biz": null,
"monitor": null,
"l1": null,
"l2": null,
"l3": null,
"summary_table_id": null
}
```

### 状态码：400

#### BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

#### Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

#### Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据下列参数，新建复合指标。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateDesignCompoundMetricSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    CreateDesignCompoundMetricRequest request = new CreateDesignCompoundMetricRequest();
    CompoundMetricVO body = new CompoundMetricVO();
    MetricMonitorVO monitorbody = new MetricMonitorVO();
    monitorbody.withExpression("")
        .withMetricType(MetricMonitorVO.MetricTypeEnum.fromValue("COMPOUND_METRIC"));
    List<String> listbodyMetricIds = new ArrayList<>();
    listbodyMetricIds.add("1300493917942329344");
    listbodyMetricIds.add("1268168163686780928");
    listbodyMetricIds.add("1181167466668310528");
    body.withMonitor(monitorbody);
    body.withDataType("STRING");
    body.withL3Id("1169309252771299328");
    body.withCalExp("${1300493917942329344}+${1268168163686780928}+${1181167466668310528}");
    body.withMetricIds(listbodyMetricIds);
    body.withCompoundType(CompoundMetricVO.CompoundTypeEnum.fromValue("EXPRESSION"));
    body.withDimensionGroup("1169322847009742848");
    body.withDescription("");
    body.withNameCh("测试复合指标");
    body.withNameEn("test_compound_index");
    body.withId("0");
    request.withBody(body);
    try {
        CreateDesignCompoundMetricResponse response = client.createDesignCompoundMetric(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

根据下列参数，新建复合指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateDesignCompoundMetricRequest()
        monitorbody = MetricMonitorVO(
            expression="",
            metric_type="COMPOUND_METRIC"
        )
        listMetricIdsbody = [
            "1300493917942329344",
            "1268168163686780928",
            "1181167466668310528"
        ]
        request.body = CompoundMetricVO(
            monitor=monitorbody,
            data_type="STRING",
            l3_id="1169309252771299328",
            cal_exp="${1300493917942329344}+${1268168163686780928}+${1181167466668310528}",
            metric_ids=listMetricIdsbody,
            compound_type="EXPRESSION",
            dimension_group="1169322847009742848",
            description="",
            name_ch="测试复合指标",
            name_en="test_compound_index",
            id="0"
        )
        response = client.create_design_compound_metric(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据下列参数，新建复合指标。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```

ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateDesignCompoundMetricRequest{
    expressionMonitor:= ""
    metricTypeMonitor:= model.GetMetricMonitorVoMetricTypeEnum().COMPOUND_METRIC
    monitorbody := &model.MetricMonitorVo{
        Expression: &expressionMonitor,
        MetricType: &metricTypeMonitor,
    }
    var listMetricIdsbody = []string{
        "1300493917942329344",
        "1268168163686780928",
        "1181167466668310528",
    }
    calExpCompoundMetricVo:= "${1300493917942329344}+${1268168163686780928}+
${1181167466668310528}"
    descriptionCompoundMetricVo:= ""
    idCompoundMetricVo:= "0"
    request.Body = &model.CompoundMetricVo{
        Monitor: monitorbody,
        DataType: "STRING",
        L3Id: "1169309252771299328",
        CalExp: &calExpCompoundMetricVo,
        MetricIds: &listMetricIdsbody,
        CompoundType: model.GetCompoundMetricVoCompoundTypeEnum().EXPRESSION,
        DimensionGroup: "1169322847009742848",
        Description: &descriptionCompoundMetricVo,
        NameCh: "测试复合指标",
        NameEn: "test_compound_index",
        Id: &idCompoundMetricVo,
    }
}
response, err := client.CreateDesignCompoundMetric(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 CompoundMetricVO。
400	BadRequest

状态码	描述
401	Unauthorized
403	Forbidden

### 8.14.3 更新复合指标

#### 功能介绍

根据参数，更新复合指标。

#### 调用方法

请参见[如何调用API](#)。

#### URI

PUT /v2/{project\_id}/design/compound-metrics

表 8-786 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 8-787 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

表 8-788 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	复合指标ID, ID字符串。更新时必填。
name_en	是	String	复合指标英文名称。
name_ch	是	String	复合指标名称。
description	否	String	描述
dimension_group	是	String	统计维度ID, 更新时不可修改此值。
compound_type	是	String	复合指标类型。 枚举值: <ul style="list-style-type: none"> <li>• EXPRESSION: 表达式</li> <li>• PERIODICITY_VALUED_COMPARISON: 环比增长率</li> <li>• INTERVAL_VALUED_COMPARISON: 同比增长率</li> </ul>
comparison_type	否	String	同比配置, 复合指标类型为“同比增长率”时必填。 枚举值: <ul style="list-style-type: none"> <li>• YEAR_TO_YEAR: 年同比</li> <li>• MONTH_TO_MONTH: 月同比</li> <li>• WEEK_TO_WEEK: 周同比</li> </ul>
metric_ids	否	Array of strings	设定衍生指标, 复合指标类型为“同比增长率”或“环比增长率”时必填, ID字符串。
metric_names	否	Array of strings	指标名称信息。
compound_metric_ids	否	Array of strings	复合指标信息, ID字符串。

参数	是否必选	参数类型	描述
compound_metric_names	否	Array of strings	复合指标名称信息。
cal_fn_ids	否	Array of strings	引用函数ID, ID字符串。
cal_exp	否	String	计算表达式, 复合指标类型为“表达式”时必填。形如\${index_id} + \${compound#index_id}, 其中index_id代表引用的衍生指标ID, compound#index_id代表引用的复合指标ID。
l2_id	否	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	是	String	业务对象ID, ID字符串。
data_type	是	String	字段类型。
create_by	否	String	创建人。
update_by	否	String	更新人。
status	否	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManageVO object	业务版本管理, 只读。
monitor	否	MetricMonitorVO object	指标监控信息。
l1	否	String	主题域分组中文名, 只读, 创建和更新时无需填写。

参数	是否必选	参数类型	描述
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。

表 8-789 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。



参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-790 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-791 MetricMonitorVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
other_metric_ids	否	Array of strings	其他指标ID，ID字符串。
other_compond_metric_ids	否	Array of strings	其他复合指标ID。
expression	否	String	告警表达式。
metric_id	否	String	挂载指ID，ID字符串。
front_configs	否	String	前端表达式配置，用于前端数据恢复。

参数	是否必选	参数类型	描述
metric_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>



参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

## 响应参数

状态码：200

表 8-792 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	更新复合指标的返回结果，成功的个数。

表 8-793 data

参数	参数类型	描述
value	<b>CompoundMetricVO</b> object	更新复合指标操作返回结果。

表 8-794 CompoundMetricVO

参数	参数类型	描述
id	String	复合指标ID, ID字符串。更新时必填。
name_en	String	复合指标英文名称。
name_ch	String	复合指标名称。
description	String	描述
dimension_group	String	统计维度ID, 更新时不可修改此值。
group_name	String	颗粒度名称, 只读。
group_code	String	颗粒度编码, 只读。
compound_type	String	复合指标类型。 枚举值: <ul style="list-style-type: none"> <li>• EXPRESSION: 表达式</li> <li>• PERIODICITY_VALUED_COMPARISON: 环比增长率</li> <li>• INTERVAL_VALUED_COMPARISON: 同比增长率</li> </ul>
comparison_type	String	同比配置, 复合指标类型为“同比增长率”时必填。 枚举值: <ul style="list-style-type: none"> <li>• YEAR_TO_YEAR: 年同比</li> <li>• MONTH_TO_MONTH: 月同比</li> <li>• WEEK_TO_WEEK: 周同比</li> </ul>
metric_ids	Array of strings	设定衍生指标, 复合指标类型为“同比增长率”或“环比增长率”时必填, ID字符串。
metric_names	Array of strings	指标名称信息。
compound_metric_ids	Array of strings	复合指标信息, ID字符串。
compound_metric_names	Array of strings	复合指标名称信息。
cal_fn_ids	Array of strings	引用函数ID, ID字符串。

参数	参数类型	描述
cal_exp	String	计算表达式，复合指标类型为“表达式”时必填。形如\${index_id} + \${compound#index_id}，其中index_id代表引用的衍生指标ID，compound#index_id代表引用的复合指标ID。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
data_type	String	字段类型。
create_by	String	创建人。
update_by	String	更新人。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
monitor	<b>MetricMonitorVO</b> object	指标监控信息。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
summary_table_id	String	汇总表ID，只读，ID字符串。

表 8-795 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-796 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-797 MetricMonitorVO

参数	参数类型	描述
id	String	编码，ID字符串。
other_metric_ids	Array of strings	其他指标ID，ID字符串。
other_metric_names	Array of strings	其他指标名称，只读。
other_compound_metric_ids	Array of strings	其他复合指标ID。
other_compound_metric_names	Array of strings	其他复合指标名称。
expression	String	告警表达式。
metric_id	String	挂载指ID，ID字符串。
front_configs	String	前端表达式配置，用于前端数据恢复。

参数	参数类型	描述
metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

**状态码：400**

表 8-798 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-799 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-800 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据下列参数，更新复合指标。

PUT https://{endpoint}/v2/{project\_id}/design/compound-metrics

```
{
  "id": "1300537267533774848",
  "name_ch": "测试复合指标",
  "name_en": "test_compound_index",
  "l3_id": "1169309252771299328",
  "dimension_group": "1169322847009742848",
  "data_type": "STRING",
  "compound_type": "EXPRESSION",
  "monitor": {
    "id": null,
    "other_metric_ids": [ ],
    "other_compound_metric_ids": [ ],
    "expression": "",
    "metric_id": "1300537267533774848",
    "metric_type": "COMPOUND_METRIC"
  },
  "description": "",
  "metric_ids": [ "1300493917942329344", "1268168163686780928", "1181167466668310528" ],
  "compound_metric_ids": [ ],
  "cal_exp": "${1300493917942329344}+${1268168163686780928}+${1181167466668310528}"
}
```

## 响应示例

**状态码：200**

Success，返回data数据是 CompoundMetricVO。

```
{
  "value": {
    "id": "1300537267533774848",
    "name_en": "test_compound_index",
    "name_ch": "测试复合指标",
    "description": "",
    "dimension_group": "1169322847009742848",
    "group_name": null,
    "group_code": null,
    "compound_type": "EXPRESSION",
    "comparison_type": null,
    "metric_ids": [ "1300493917942329344", "1268168163686780928", "1181167466668310528" ],
    "metric_names": null,
    "compound_metric_ids": [ ],
    "compound_metric_names": null,
    "cal_fn_ids": null,
    "cal_exp": "${1300493917942329344}+${1268168163686780928}+${1181167466668310528}",
    "l1_id": null,
    "l2_id": null,
    "l3_id": "1169309252771299328",
  }
}
```

```
"data_type": "STRING",
"create_by": "test_user",
"update_by": "test_user",
"status": "DRAFT",
"create_time": "2024-10-28T19:10:47+08:00",
"update_time": "2024-10-28T19:20:23.085+08:00",
"approval_info": null,
"new_biz": null,
"monitor": null,
"l1": null,
"l2": null,
"l3": null,
"summary_table_id": null
}
}
```

### 状态码：400

#### BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

#### Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

#### Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据下列参数，更新复合指标。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateDesignCompoundMetricSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    UpdateDesignCompoundMetricRequest request = new UpdateDesignCompoundMetricRequest();
    CompoundMetricVO body = new CompoundMetricVO();
    MetricMonitorVO monitorbody = new MetricMonitorVO();
    monitorbody.withExpression("")
        .withMetricId("1300537267533774848")
        .withMetricType(MetricMonitorVO.MetricTypeEnum.fromValue("COMPOUND_METRIC"));
    List<String> listbodyMetricIds = new ArrayList<>();
    listbodyMetricIds.add("1300493917942329344");
    listbodyMetricIds.add("1268168163686780928");
    listbodyMetricIds.add("1181167466668310528");
    body.withMonitor(monitorbody);
    body.withDataType("STRING");
    body.withL3Id("1169309252771299328");
    body.withCalExp("${1300493917942329344}+${1268168163686780928}+${1181167466668310528}");
    body.withMetricIds(listbodyMetricIds);
    body.withCompoundType(CompoundMetricVO.CompoundTypeEnum.fromValue("EXPRESSION"));
    body.withDimensionGroup("1169322847009742848");
    body.withDescription("");
    body.withNameCh("测试复合指标");
    body.withNameEn("test_compound_index");
    body.withId("1300537267533774848");
    request.withBody(body);
    try {
        UpdateDesignCompoundMetricResponse response = client.updateDesignCompoundMetric(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

根据下列参数，更新复合指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateDesignCompoundMetricRequest()
        monitorbody = MetricMonitorVO(
            expression="",
            metric_id="1300537267533774848",
            metric_type="COMPOUND_METRIC"
        )
        listMetricIdsbody = [
            "1300493917942329344",
            "1268168163686780928",
            "1181167466668310528"
        ]
        request.body = CompoundMetricVO(
            monitor=monitorbody,
            data_type="STRING",
            l3_id="1169309252771299328",
            cal_exp="{1300493917942329344}+{1268168163686780928}+{1181167466668310528}",
            metric_ids=listMetricIdsbody,
            compound_type="EXPRESSION",
            dimension_group="1169322847009742848",
            description="",
            name_ch="测试复合指标",
            name_en="test_compound_index",
            id="1300537267533774848"
        )
        response = client.update_design_compound_metric(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

根据下列参数，更新复合指标。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
```



```
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateDesignCompoundMetricRequest{
    expressionMonitor:= ""
    metricIdMonitor:= "1300537267533774848"
    metricTypeMonitor:= model.GetMetricMonitorVoMetricTypeEnum().COMPOUND_METRIC
    monitorbody := &model.MetricMonitorVo{
        Expression: &expressionMonitor,
        MetricId: &metricIdMonitor,
        MetricType: &metricTypeMonitor,
    }
    var listMetricIdsbody = []string{
        "1300493917942329344",
        "1268168163686780928",
        "1181167466668310528",
    }
    calExpCompoundMetricVo:= "${1300493917942329344}+${1268168163686780928}+
${1181167466668310528}"
    descriptionCompoundMetricVo:= ""
    idCompoundMetricVo:= "1300537267533774848"
    request.Body = &model.CompoundMetricVo{
        Monitor: monitorbody,
        DataType: "STRING",
        L3Id: "1169309252771299328",
        CalExp: &calExpCompoundMetricVo,
        MetricIds: &listMetricIdsbody,
        CompoundType: model.GetCompoundMetricVoCompoundTypeEnum().EXPRESSION,
        DimensionGroup: "1169322847009742848",
        Description: &descriptionCompoundMetricVo,
        NameCh: "测试复合指标",
        NameEn: "test_compound_index",
        Id: &idCompoundMetricVo,
    }
}
response, err := client.UpdateDesignCompoundMetric(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 CompoundMetricVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.14.4 删除复合指标

### 功能介绍

根据ID集合删除复合指标。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/compound-metrics

表 8-801 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-802 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 (响应消息头中X-Subject-Token的值) 使用Token认证时必选。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-803 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-804 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-805 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-806 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-807 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-808 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入的参数，删除复合指标，只支持删除状态为草稿、已下线、已驳回的复合指标。

```
DELETE https://{endpoint}/v2/{project_id}/design/compound-metrics
{
  "ids" : [ "1227990857618227200" ]
}
```

## 响应示例

状态码：200

Success，返回data数据是删除个数。

```
{  
  "value" : 1  
}
```

**状态码：400**

BadRequest

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user request is illegal."  
}
```

**状态码：401**

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

**状态码：403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据传入的参数，删除复合指标，只支持删除状态为草稿、已下线、已驳回的复合指标。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class DeleteDesignCompoundMetricSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteDesignCompoundMetricRequest request = new DeleteDesignCompoundMetricRequest();
IdsParam body = new IdsParam();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1227990857618227200");
body.withIds(listbodyIds);
request.withBody(body);
try {
    DeleteDesignCompoundMetricResponse response = client.deleteDesignCompoundMetric(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据传入的参数，删除复合指标，只支持删除状态为草稿、已下线、已驳回的复合指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteDesignCompoundMetricRequest()
        listbody = [
            "1227990857618227200"
        ]
        request.body = IdsParam(
            ids=listbody
        )
```

```
response = client.delete_design_compound_metric(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据传入的参数，删除复合指标，只支持删除状态为草稿、已下线、已驳回的复合指标。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDesignCompoundMetricRequest{}
    var listIdsbody = []string{
        "1227990857618227200",
    }
    request.Body = &model.IdsParam{
        Ids: listIdsbody,
    }
    response, err := client.DeleteDesignCompoundMetric(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.14.5 查看复合指标详情

### 功能介绍

通过ID获取复合指标详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/compound-metrics/{id}

表 8-809 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID, ID字符串。

表 8-810 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。



## 请求参数

表 8-811 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-812 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	返回数据。

表 8-813 data

参数	参数类型	描述
value	<a href="#">CompoundMetricVO</a> object	返回的数据信息。

表 8-814 CompoundMetricVO

参数	参数类型	描述
id	String	复合指标ID，ID字符串。更新时必填。

参数	参数类型	描述
name_en	String	复合指标英文名称。
name_ch	String	复合指标名称。
description	String	描述
dimension_group	String	统计维度ID, 更新时不可修改此值。
group_name	String	颗粒度名称, 只读。
group_code	String	颗粒度编码, 只读。
compound_type	String	复合指标类型。 枚举值: <ul style="list-style-type: none"> <li>• EXPRESSION: 表达式</li> <li>• PERIODICITY_VALUED_COMPARISON: 环比增长率</li> <li>• INTERVAL_VALUED_COMPARISON: 同比增长率</li> </ul>
comparison_type	String	同比配置, 复合指标类型为“同比增长率”时必填。 枚举值: <ul style="list-style-type: none"> <li>• YEAR_TO_YEAR: 年同比</li> <li>• MONTH_TO_MONTH: 月同比</li> <li>• WEEK_TO_WEEK: 周同比</li> </ul>
metric_ids	Array of strings	设定衍生指标, 复合指标类型为“同比增长率”或“环比增长率”时必填, ID字符串。
metric_names	Array of strings	指标名称信息。
compound_metric_ids	Array of strings	复合指标信息, ID字符串。
compound_metric_names	Array of strings	复合指标名称信息。
cal_fn_ids	Array of strings	引用函数ID, ID字符串。
cal_exp	String	计算表达式, 复合指标类型为“表达式”时必填。形如\${index_id} + \${compound#index_id}, 其中index_id代表引用的衍生指标ID, compound#index_id代表引用的复合指标ID。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。

参数	参数类型	描述
l3_id	String	业务对象ID, ID字符串。
data_type	String	字段类型。
create_by	String	创建人。
update_by	String	更新人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理, 只读。
monitor	<b>MetricMonitorV</b> <b>O</b> object	指标监控信息。
l1	String	主题域分组中文名, 只读, 创建和更新时无需填写。
l2	String	主题域中文名, 只读, 创建和更新时无需填写。
l3	String	业务对象中文名, 只读, 创建和更新时无需填写。
summary_table_id	String	汇总表ID, 只读, ID字符串。

表 8-815 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-816 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-817 MetricMonitorVO

参数	参数类型	描述
id	String	编码，ID字符串。
other_metric_ids	Array of strings	其他指标ID，ID字符串。
other_metric_names	Array of strings	其他指标名称，只读。
other_compound_metric_ids	Array of strings	其他复合指标ID。
other_compound_metric_names	Array of strings	其他复合指标名称。
expression	String	告警表达式。
metric_id	String	挂载指ID，ID字符串。
front_configs	String	前端表达式配置，用于前端数据恢复。

参数	参数类型	描述
metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>

**状态码：400**

表 8-818 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-819 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-820 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据复合指标ID，查询复合指标详情。

GET https://{endpoint}/v2/{project\_id}/design/compound-metrics/1170056628536553472

## 响应示例

状态码：200

Success，返回data数据是CompoundMetricVO详情。

```
{
  "data": {
    "value": {
      "id": "1170056628536553472",
      "name_en": "com_compound_metric_test",
      "name_ch": "复合指标测试",
      "description": "",
      "dimension_group": "1169322847009742848,1169329693619625984.1169329693695123460.1169327565677182976",
      "group_name": "供应商,行程订单.下车时间",
      "group_code": "dim_vendor,fact_stroke_order.dim_dropoff",
      "compound_type": "EXPRESSION",
      "comparison_type": null,
      "metric_ids": [ "1170007051066355712" ],
      "metric_names": [ "总车费(供应商,行程订单.下车时间)" ],
      "compound_metric_ids": [ ],
      "compound_metric_names": null,
      "cal_fn_ids": null,
      "cal_exp": "${1170007051066355712}/2",
      "l1_id": "1169309252771299328",
      "l2_id": "1169309374670356480,1169310587398889472",
      "l3_id": null,
      "data_type": "BIGINT",
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "status": "PUBLISHED",
      "create_time": "2023-11-03T17:47:19+08:00",
      "update_time": "2023-11-03T17:55:12+08:00",
      "approval_info": {
        "id": "1170056629291528192",
        "tenant_id": null,
        "name_ch": null,
        "name_en": null,
        "biz_id": "1170056628536553472",
        "biz_type": null,
        "biz_info": null,
        "biz_info_obj": null,
        "biz_version": 0,
        "biz_status": "PUBLISHED",
        "approval_status": "APPROVED",
        "approval_type": null,
        "submit_time": "2023-11-03T17:47:19+08:00",
      }
    }
  }
}
```

```
"create_by" : null,
"l1" : null,
"l2" : null,
"l3" : null,
"approval_time" : "2023-11-03T17:55:12+08:00",
"approver" : "test_uesr",
"email" : null,
"msg" : "Fast Approval OK.",
"directory_path" : null
},
"new_biz" : null,
"monitor" : {
  "id" : "1170056628612050944",
  "other_metric_ids" : [ "1170007051066355712" ],
  "other_metric_names" : [ "总车费(供应商,行程订单,下车时间)" ],
  "other_compound_metric_ids" : [ ],
  "other_compound_metric_names" : null,
  "expression" : "{$1170007051066355712}/2 < 0",
  "metric_id" : "1170056628536553472",
  "front_configs" : null,
  "metric_type" : "COMPOUND_METRIC"
},
"l1" : "城市交通",
"l2" : "行程记录/行程事实",
"l3" : null,
"summary_table_id" : "1170007052639219712"
}
}
```

#### 状态码：400

##### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

##### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

##### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowCompoundMetricByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowCompoundMetricByIdRequest request = new ShowCompoundMetricByIdRequest();
        request.withId("{id}");
        try {
            ShowCompoundMetricByIdResponse response = client.showCompoundMetricById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowCompoundMetricByIdRequest()
    request.id = "{id}"
    response = client.show_compound_metric_by_id(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowCompoundMetricByIdRequest{}
    request.Id = "{id}"
    response, err := client.ShowCompoundMetricById(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。



## 状态码

状态码	描述
200	Success, 返回data数据是CompoundMetricVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.15 维度接口

### 8.15.1 查找维度

#### 功能介绍

通过中英文名称、描述、创建者、审核人、状态、l3Id、衍生指标idList、修改时间分页查找维度信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/dimensions

表 8-821 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-822 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
name_ch	否	String	按名称精确查询。
name_en	否	String	按编码精确查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。

参数	是否必选	参数类型	描述
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
l2_id	否	String	主题域l2的ID, ID字符串。
derivative_ids	否	Array of strings	依据复合指标ID列表查维度, ID字符串。
begin_time	否	String	时间过滤左边界, 与end_time一起使用, 只支持时间范围过滤, 单边过滤无效。格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界, 与begin_time一起使用只支持时间范围过滤, 单边过滤无效。格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
fact_logic_id	否	String	事实表ID, ID字符串。
dimension_type	否	String	维度类型。 枚举值： <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
limit	否	Integer	每页查询条数, 即查询Y条数据。默认值50, 取值范围[1,100]。
offset	否	Integer	查询起始坐标, 即跳过X条数据, 仅支持0或limit的整数倍, 不满足则向下取整, 默认值0。
biz_catalog_id	否	String	所属的业务分层的ID。

## 请求参数

表 8-823 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-824 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-825 data

参数	参数类型	描述
value	<b>value</b> object	value，统一的返回结果的外层数据结构。

表 8-826 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>DimensionVO</b> objects	DimensionVO信息。

表 8-827 DimensionVO

参数	参数类型	描述
id	String	编码（新建时只读，更新时必须填），ID字符串。
name_en	String	维度英文名称。
dimension_type	String	维度类型（更新时只能由普通维度改为层级维度，其余场景都不允许修改）。 枚举值： <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
name_ch	String	维度名称。
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
code_table_id	String	引用码表ID（新建/更新“码表维度”时必须填），ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表，只读。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
hierarchies	Array of <b>DimensionHierarchiesVO</b> objects	层级属性。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	<b>BizDatasourceRelationVO</b> object	数据源信息，新建/更新时必填。
owner	String	资产责任人，新建/更新时必填。
obs_location	String	外表路径
table_type	String	表类型。

参数	参数类型	描述
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	String	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-828 CodeTableVO

参数	参数类型	描述
id	String	码表ID，ID字符串。
name_en	String	表名称，英文名。
name_ch	String	表名称，中文名。

参数	参数类型	描述
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-829 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必须填), ID字符串。

参数	参数类型	描述
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-830 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-831 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码, ID字符串。
name	String	层级名称。



参数	参数类型	描述
attrs	Array of <b>HierarchiesAttrVO</b> objects	层级包含的属性。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-832 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码，ID字符串。
hierarchies_id	String	层级ID，ID字符串。
attr_id	String	属性ID，ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称，只读。
detail_attr_ids	Array of strings	详情属性ID，ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文，只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段，只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段，只读。

表 8-833 DimensionAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
dimension_id	String	维度ID，只读，ID字符串。
code_table_field_id	String	码表属性ID，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。

表 8-834 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-835 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-836 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。

表 8-837 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-838 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-839 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-840 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-841 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-842 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-843 BizDatasourceRelationVO

参数	参数类型	描述
id	String	编码, ID字符串。

参数	参数类型	描述
biz_id	String	业务对象信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	String	数据连接ID。
dw_name	String	数据连接名，只读。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。

表 8-844 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-845 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

**状态码：400**

**表 8-846** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-847** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-848** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据查询条件，分页查询维度列表。

```
GET https://{endpoint}/v2/{project_id}/design/dimensions?offset=1&limit=10
```

## 响应示例

**状态码：200**

Success，返回data数据是DimensionVO数组，以及总数。

```
{  
  "data" : {
```

```

"value": {
  "total": 1,
  "records": [ {
    "id": "1222930977286987776",
    "name_en": "dim_test_czh_0328",
    "dimension_type": "COMMON",
    "name_ch": "test_czh_0328",
    "description": "无",
    "create_by": "test_uesr",
    "update_by": null,
    "code_table_id": null,
    "code_table": null,
    "l1_id": "1194966710038474752",
    "l2_id": null,
    "l3_id": null,
    "hierarchies": [ ],
    "status": "PUBLISHED",
    "create_time": "2024-03-28T15:31:06+08:00",
    "update_time": "2024-03-28T15:31:06+08:00",
    "approval_info": {
      "id": "1222930978226511872",
      "tenant_id": null,
      "name_ch": null,
      "name_en": null,
      "biz_id": "1222930977286987776",
      "biz_type": null,
      "biz_info": null,
      "biz_info_obj": null,
      "biz_version": 0,
      "biz_status": "PUBLISHED",
      "approval_status": "APPROVED",
      "approval_type": null,
      "submit_time": "2024-03-28T15:31:06+08:00",
      "create_by": null,
      "l1": null,
      "l2": null,
      "l3": null,
      "approval_time": "2024-03-28T15:31:06+08:00",
      "approver": "test_uesr",
      "email": null,
      "msg": "Fast Approval OK.",
      "directory_path": null
    },
    "new_biz": null,
    "l1": "test_czh_catalog",
    "l2": null,
    "l3": null,
    "attributes": [ {
      "id": "1222930977370873856",
      "dimension_id": "1222930977286987776",
      "code_table_field_id": null,
      "name_en": "dim_test_czh_0328_key",
      "name_ch": "test_czh_0328",
      "description": "",
      "create_by": "test_uesr",
      "data_type": "BIGINT",
      "domain_type": "NUMBER",
      "data_type_extend": null,
      "is_primary_key": true,
      "is_biz_primary": false,
      "is_partition_key": false,
      "ordinal": 1,
      "not_null": false,
      "stand_row_id": null,
      "stand_row_name": null,
      "quality_infos": null,
      "secrecy_levels": [ ],
      "status": null,
      "create_time": "2024-03-28T15:31:06+08:00",
    }
  ]
}

```

```
"update_time": "2024-03-28T15:31:06+08:00",
"alias": "",
"self_defined_fields": [ {
  "fd_name_ch": "aa",
  "fd_name_en": "aa",
  "not_null": false,
  "fd_value": ""
}, {
  "fd_name_ch": "bb",
  "fd_name_en": "bb",
  "not_null": false,
  "fd_value": ""
}
], {
  "id": "1222930977370873857",
  "dimension_id": "1222930977286987776",
  "code_table_field_id": null,
  "name_en": "aaa",
  "name_ch": "aaa",
  "description": "",
  "create_by": "test_uesr",
  "data_type": "BIGINT",
  "domain_type": "NUMBER",
  "data_type_extend": null,
  "is_primary_key": false,
  "is_biz_primary": false,
  "is_partition_key": false,
  "ordinal": 2,
  "not_null": false,
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": null,
  "secrecy_levels": [ ],
  "status": null,
  "create_time": "2024-03-28T15:31:06+08:00",
  "update_time": "2024-03-28T15:31:06+08:00",
  "alias": "",
  "self_defined_fields": [ {
    "fd_name_ch": "aa",
    "fd_name_en": "aa",
    "not_null": false,
    "fd_value": ""
  }, {
    "fd_name_ch": "bb",
    "fd_name_en": "bb",
    "not_null": false,
    "fd_value": ""
  }
]
}, {
  "id": "1222930977370873858",
  "dimension_id": "1222930977286987776",
  "code_table_field_id": null,
  "name_en": "bbb",
  "name_ch": "bbb",
  "description": "",
  "create_by": "test_uesr",
  "data_type": "BIGINT",
  "domain_type": "NUMBER",
  "data_type_extend": null,
  "is_primary_key": false,
  "is_biz_primary": false,
  "is_partition_key": false,
  "ordinal": 3,
  "not_null": false,
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": null,
  "secrecy_levels": [ ],
  "status": null,
```

```
"create_time" : "2024-03-28T15:31:06+08:00",
"update_time" : "2024-03-28T15:31:06+08:00",
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : ""
}
],
"mappings" : null,
"datasource" : {
  "id" : "1222930977387651072",
  "biz_id" : "1222930977286987776",
  "biz_type" : "DIMENSION",
  "dw_type" : "DWS",
  "dw_id" : "8259af52bd294f98920ebce75a426391",
  "dw_name" : null,
  "db_name" : "bi",
  "queue_name" : "default",
  "schema" : ""
},
"owner" : "afcaca",
"table_type" : "MANAGED",
"distribute" : null,
"distribute_column" : null,
"obs_location" : "",
"alias" : "",
"configs" : "{}",
"self_defined_fields" : [ {
  "fd_name_ch" : "自定义项1",
  "fd_name_en" : "selfDefine1",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "自定义项2",
  "fd_name_en" : "selfDefine2",
  "not_null" : true,
  "fd_value" : "adada"
}
],
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
}
}
}
```

**状态码: 400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码: 401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
```

```
"error_msg" : "User authentication failed."  
}
```

**状态码: 403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## 状态码

状态码	描述
200	Success, 返回data数据是DimensionVO数组, 以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.15.2 新建维度

### 功能介绍

根据参数新建维度。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/dimensions

表 8-849 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-850 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-851 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码（新建时只读，更新时必填），ID字符串。
name_en	是	String	维度英文名称。
dimension_type	是	String	维度类型（更新时只能由普通维度改为层级维度，其余场景都不允许修改）。 枚举值： <ul style="list-style-type: none"> <li>COMMON: 普通维度</li> <li>LOOKUP: 码表维度</li> <li>HIERARCHIES: 层级维度</li> </ul>
name_ch	是	String	维度名称。
description	是	String	描述。
create_by	否	String	创建人。
update_by	否	String	更新人。



参数	是否必选	参数类型	描述
code_table_id	否	String	引用码表ID（新建/更新“码表维度”时必填），ID字符串。
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。
l3_id	是	String	业务对象ID，ID字符串。
hierarchies	否	Array of <b>DimensionHierarchiesVO</b> objects	层级属性。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	否	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	是	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	否	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	是	<b>BizDataSourceRelationVO</b> object	数据源信息，新建/更新时必填。

参数	是否必选	参数类型	描述
owner	是	String	资产责任人，新建/更新时必须填。
obs_location	否	String	外表路径
table_type	否	String	表类型。
distribute	否	String	DISTRIBUTE BY [HASH(column)  REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行 Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	否	String	DISTRIBUTE BY HASH column.
alias	否	String	别名。
self_defined_fields	否	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。
configs	否	String	其他配置
model_id	否	String	所属模型ID
model	否	<a href="#">WorkspaceVO</a> object	模型信息。

表 8-852 DimensionHierarchiesVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name	否	String	层级名称。
attrs	否	Array of <a href="#">HierarchiesAttrVO</a> objects	层级包含的属性。
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-853 HierarchiesAttrVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
hierarchies_id	否	String	层级ID，ID字符串。
attr_id	否	String	属性ID，ID字符串。
level	否	Integer	层次。
attr_name_en	否	String	引用属性编码。
detail_attr_ids	否	Array of strings	详情属性ID，ID字符串。
detail_attr_name_ens	否	Array of strings	详情属性英文。

表 8-854 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-855 DimensionAttributeVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
code_table_fiel d_id	否	String	码表属性ID，ID字符串。
name_en	是	String	字段名。
name_ch	是	String	业务属性。
description	否	String	描述。
create_by	否	String	创建人。
data_type	是	String	字段类型。
data_type_ext end	否	String	数据类型扩展字段。
is_primary_ke y	是	Boolean	是否主键。
is_biz_primary	否	Boolean	是否业务主键。
is_partition_k ey	否	Boolean	是否分区。
ordinal	是	Integer	序号。
not_null	否	Boolean	是否不为空。
stand_row_id	否	String	关联的数据标准的ID，ID字符串。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
alias	否	String	别名
self_defined_fi elds	否	Array of <b>SelfDefinedF ieldVO</b> objects	自定义项。

表 8-856 TableMappingVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name	是	String	名称。
description	否	String	描述。
target_table_id	否	String	目的表ID，ID字符串。
src_model_id	否	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	否	String	来源模型名称。
view_text	否	String	采集的视图来源，dws视图逆向使用。
target_table_name	否	String	目的表名称。
details	否	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	否	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	否	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	否	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	否	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	否	String	创建人。
update_by	否	String	更新人。



表 8-857 TableMappingDetailVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
mapping_id	否	String	名称。
target_attr_id	否	String	目的字段ID，ID字符串。
target_attr_name	是	String	目的字段排序。
src_table_ids	否	String	源表ID。
src_attr_ids	否	String	源表字段ID。
remark	否	String	备注。
create_time	否	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	否	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	否	String	创建人。
update_by	否	String	更新人。
target_attr	否	Object	目标属性。

表 8-858 MappingSourceTableVO

参数	是否必选	参数类型	描述
table1_id	是	String	表1ID，ID字符串。
table2_id	否	String	表2ID，ID字符串。
table1_name	是	String	表1名称。
table2_name	否	String	表2名称。

参数	是否必选	参数类型	描述
join_type	是	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	是	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-859 MappingJoinFieldVO

参数	是否必选	参数类型	描述
field1_id	是	String	属性1ID, ID字符串。
field2_id	是	String	属性2ID, ID字符串。
field1_name	是	String	属性1名称。
field2_name	是	String	属性2名称。

表 8-860 MappingSourceFieldVO

参数	是否必选	参数类型	描述
target_field_id	否	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	是	String	目标字段编码。
field_ids	否	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	否	String	转换表达式。
field_names	否	Array of strings	来源字段名称列表。
changed	否	Boolean	字段是否发生变化。

表 8-861 BizDatasourceRelationVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
biz_id	否	String	业务对象信息，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	是	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	是	String	数据连接ID。
db_name	否	String	数据库名。
queue_name	否	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	否	String	DWS类型需要。

表 8-862 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

表 8-863 WorkspaceVO

参数	是否必选	参数类型	描述
id	否	String	编号，ID字符串。
name	是	String	工作区名字。
description	否	String	描述。
is_physical	否	Boolean	是否为物理表。
frequent	否	Boolean	是否为常用。
top	否	Boolean	分层治理。
level	否	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	否	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_by	否	String	创建人。
update_by	否	String	更新人。
type	是	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	否	String	关联的业务分层的ID列表 { "l1Ids": [], "l2Ids": [], "l3Ids": [] }。

参数	是否必选	参数类型	描述
databases	否	Array of strings	数据库名称数组。
table_model_prefix	否	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

## 响应参数

状态码：200

表 8-864 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-865 data

参数	参数类型	描述
value	<b>DimensionVO</b> object	value，统一的返回结果的外层数据结构。

表 8-866 DimensionVO

参数	参数类型	描述
id	String	编码（新建时只读，更新时必填），ID 字符串。
name_en	String	维度英文名称。
dimension_type	String	维度类型（更新时只能由普通维度改为层级维度，其余场景都不允许修改）。 枚举值： <ul style="list-style-type: none"> <li>COMMON: 普通维度</li> <li>LOOKUP: 码表维度</li> <li>HIERARCHIES: 层级维度</li> </ul>
name_ch	String	维度名称。
description	String	描述。
create_by	String	创建人。

参数	参数类型	描述
update_by	String	更新人。
code_table_id	String	引用码表ID（新建/更新“码表维度”时必填），ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表，只读。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
hierarchies	Array of <b>DimensionHierarchiesVO</b> objects	层级属性。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。



参数	参数类型	描述
l3	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	<b>BizDatasourceRelationVO</b> object	数据源信息，新建/更新时必填。
owner	String	资产责任人，新建/更新时必填。
obs_location	String	外表路径
table_type	String	表类型。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。

参数	参数类型	描述
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	String	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-867 CodeTableVO

参数	参数类型	描述
id	String	码表ID, ID字符串。
name_en	String	表名称, 英文名。
name_ch	String	表名称, 中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>DRAFT: 草稿</li> <li>PUBLISH_DEVELOPING: 发布待审核</li> <li>PUBLISHED: 已发布</li> <li>OFFLINE_DEVELOPING: 下线待审核</li> <li>OFFLINE: 已下线</li> <li>REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-868 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <b>CodeTableFieldValueVO</b> objects	码表属性值。

参数	参数类型	描述
count_field_values	Integer	码表属性值总数。

表 8-869 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-870 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码, ID字符串。
name	String	层级名称。
attrs	Array of <a href="#">HierarchiesAttrVO</a> objects	层级包含的属性。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-871 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码, ID字符串。
hierarchies_id	String	层级ID, ID字符串。

参数	参数类型	描述
attr_id	String	属性ID, ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称, 只读。
detail_attr_ids	Array of strings	详情属性ID, ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文, 只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段, 只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段, 只读。

表 8-872 DimensionAttributeVO

参数	参数类型	描述
id	String	编码, ID字符串。
dimension_id	String	维度ID, 只读, ID字符串。
code_table_field_id	String	码表属性ID, ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>

参数	参数类型	描述
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID, ID字符串。
stand_row_name	String	关联的数据标准名称, 只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息, 只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。

表 8-873 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-874 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-875 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。

表 8-876 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-877 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-878 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。



参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-879 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>LEFT: 左外连接</li> <li>RIGHT: 右外连接</li> <li>INNER: 内连接</li> <li>FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-880 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-881 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-882 BizDatasourceRelationVO

参数	参数类型	描述
id	String	编码, ID字符串。

参数	参数类型	描述
biz_id	String	业务对象信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	String	数据连接ID。
dw_name	String	数据连接名，只读。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。

表 8-883 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-884 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

**状态码：400**

**表 8-885** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-886** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-887** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据下列参数，新建维度。

POST https://{endpoint}/v2/{project\_id}/design/dimensions

```
{
  "id": 0,
  "model_id": "1255482244563357696",
  "name_ch": "测试维度",
  "name_en": "dim_test_dimension",
  "alias": "",
  "owner": "test_user",
  "dimension_type": "COMMON",
```

```
"description": "无",
"l3_id": "1169309252771299328",
"attributes": [ {
  "id": 0,
  "name_ch": "测试维度_代理键",
  "name_en": "dim_key",
  "alias": "",
  "description": "",
  "data_type": "BIGINT",
  "data_type_extend": null,
  "is_biz_primary": false,
  "is_primary_key": true,
  "is_partition_key": false,
  "not_null": false,
  "ordinal": 1,
  "dimension_id": 0,
  "create_by": "string",
  "stand_row_id": null,
  "stand_row_name": null,
  "secrecy_levels": null,
  "self_defined_fields": [ ]
}, {
  "id": 0,
  "name_ch": "测试维度_属性",
  "name_en": "test_field",
  "alias": "",
  "description": "",
  "data_type": "BIGINT",
  "data_type_extend": null,
  "is_biz_primary": false,
  "is_primary_key": false,
  "is_partition_key": false,
  "not_null": false,
  "ordinal": 2,
  "dimension_id": 0,
  "create_by": "string",
  "stand_row_id": null,
  "stand_row_name": null,
  "secrecy_levels": null,
  "self_defined_fields": [ ]
} ],
"datasource": {
  "id": 0,
  "biz_type": "DIMENSION",
  "dw_type": "DWS",
  "dw_id": "8259af52bd294f98920ebce75a426391",
  "db_name": "bi",
  "schema": ""
},
"obs_location": "",
"configs": "{}",
"table_type": "MANAGED",
"mappings": [ ]
}
```

## 响应示例

**状态码: 200**

Success, 返回data数据是 DimensionVO。

```
{
  "data": {
    "value": {
      "id": "1300473264673533952",
      "name_en": "dim_test_dimension",
      "dimension_type": "COMMON",
      "name_ch": "测试维度",
      "description": "无",

```



```

"create_by" : "test_uesr",
"update_by" : "test_uesr",
"code_table_id" : null,
"code_table" : null,
"l1_id" : null,
"l2_id" : null,
"l3_id" : "1169309252771299328",
"hierarchies" : null,
"status" : "DRAFT",
"create_time" : "2024-10-28T14:56:27.923+08:00",
"update_time" : "2024-10-28T14:56:27.923+08:00",
"approval_info" : null,
"new_biz" : null,
"l1" : null,
"l2" : null,
"l3" : null,
"attributes" : [ {
  "id" : "0",
  "dimension_id" : "0",
  "code_table_field_id" : null,
  "name_en" : "dim_key",
  "name_ch" : "测试维度_代理键",
  "description" : "",
  "create_by" : "string",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : null,
  "is_primary_key" : true,
  "is_biz_primary" : false,
  "is_partition_key" : false,
  "ordinal" : 1,
  "not_null" : false,
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "secrecy_levels" : null,
  "status" : null,
  "create_time" : "2024-10-28T14:56:27.923+08:00",
  "update_time" : "2024-10-28T14:56:27.923+08:00",
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "aa",
    "fd_name_en" : "aa",
    "not_null" : false,
    "fd_value" : ""
  }, {
    "fd_name_ch" : "bb",
    "fd_name_en" : "bb",
    "not_null" : false,
    "fd_value" : ""
  } ],
  "associated_code_table_field" : null
}, {
  "id" : "0",
  "dimension_id" : "0",
  "code_table_field_id" : null,
  "name_en" : "test_field",
  "name_ch" : "测试维度_属性",
  "description" : "",
  "create_by" : "string",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : null,
  "is_primary_key" : false,
  "is_biz_primary" : false,
  "is_partition_key" : false,
  "ordinal" : 2,
  "not_null" : false,
  "stand_row_id" : null,

```

```
"stand_row_name" : null,
"quality_infos" : null,
"secrecy_levels" : null,
"status" : null,
"create_time" : "2024-10-28T14:56:27.923+08:00",
"update_time" : "2024-10-28T14:56:27.923+08:00",
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : ""
} ],
"associated_code_table_field" : null
}],
"mappings" : [ ],
"datasource" : {
  "id" : "0",
  "biz_id" : null,
  "biz_type" : "DIMENSION",
  "dw_type" : "DLI",
  "dw_id" : "8259af52bd294f98920ebce75a426391",
  "dw_name" : null,
  "db_name" : "bi",
  "queue_name" : "default",
  "schema" : ""
},
"owner" : "test_user",
"table_type" : "MANAGED",
"distribute" : null,
"distribute_column" : null,
"compression" : null,
"obs_location" : "",
"pre_combine_field" : null,
"alias" : "",
"configs" : "{}",
"self_defined_fields" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : null,
"model_id" : "1255482244563357696",
"model" : null
}
}
```

### 状态码：400

#### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

#### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**SDK 代码示例**

SDK代码示例如下。

**Java**

根据下列参数，新建维度。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateDesignDimensionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateDesignDimensionRequest request = new CreateDesignDimensionRequest();
        DimensionVO body = new DimensionVO();
        BizDatasourceRelationVO datasourcebody = new BizDatasourceRelationVO();
        datasourcebody.withId("0")
            .withBizType(BizDatasourceRelationVO.BizTypeEnum.fromValue("DIMENSION"))
            .withDwType("DWS")
            .withDwId("8259af52bd294f98920ebce75a426391")
            .withDbName("bi")
            .withSchema("");
        List<DimensionAttributeVO> listbodyAttributes = new ArrayList<>();
        listbodyAttributes.add(
            new DimensionAttributeVO()
                .withId("0")
                .withDimensionId("0")
                .withNameEn("dim_key")
        );
    }
}
```

```
.withNameCh("测试维度_代理键")
.withDescription("")
.withCreateBy("string")
.withDataType("BIGINT")
.withIsPrimaryKey(true)
.withIsBizPrimary(false)
.withIsPartitionKey(false)
.withOrdinal(1)
.withNotNull(false)
.withAlias("")
.withSelfDefinedFields()
);
listbodyAttributes.add(
    new DimensionAttributeVO()
        .withId("0")
        .withDimensionId("0")
        .withNameEn("test_field")
        .withNameCh("测试维度_属性")
        .withDescription("")
        .withCreateBy("string")
        .withDataType("BIGINT")
        .withIsPrimaryKey(false)
        .withIsBizPrimary(false)
        .withIsPartitionKey(false)
        .withOrdinal(2)
        .withNotNull(false)
        .withAlias("")
        .withSelfDefinedFields()
);
body.withConfigs("{}");
body.withAlias("");
body.withTableType("MANAGED");
body.withObsLocation("");
body.withOwner("test_user");
body.withDatasource(datasourcebody);
body.withAttributes(listbodyAttributes);
body.withL3Id("1169309252771299328");
body.withDescription("无");
body.withNameCh("测试维度");
body.withDimensionType(DimensionVO.DimensionTypeEnum.fromValue("COMMON"));
body.withNameEn("dim_test_dimension");
body.withId("0");
request.withBody(body);
try {
    CreateDesignDimensionResponse response = client.createDesignDimension(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据下列参数，新建维度。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateDesignDimensionRequest()
        datasourcebody = BizDatasourceRelationVO(
            id="0",
            biz_type="DIMENSION",
            dw_type="DWS",
            dw_id="8259af52bd294f98920ebce75a426391",
            db_name="bi",
            schema=""
        )
        listAttributesbody = [
            DimensionAttributeVO(
                id="0",
                dimension_id="0",
                name_en="dim_key",
                name_ch="测试维度_代理键",
                description="",
                create_by="string",
                data_type="BIGINT",
                is_primary_key=True,
                is_biz_primary=False,
                is_partition_key=False,
                ordinal=1,
                not_null=False,
                alias="",
            ),
            DimensionAttributeVO(
                id="0",
                dimension_id="0",
                name_en="test_field",
                name_ch="测试维度_属性",
                description="",
                create_by="string",
                data_type="BIGINT",
                is_primary_key=False,
                is_biz_primary=False,
                is_partition_key=False,
                ordinal=2,
                not_null=False,
                alias="",
            )
        ]
        request.body = DimensionVO(
            configs={},
            alias="",
            table_type="MANAGED",
            obs_location="",
            owner="test_user",
            datasource=datasourcebody,
```

```
        attributes=listAttributesbody,  
        l3_id="1169309252771299328",  
        description="无",  
        name_ch="测试维度",  
        dimension_type="COMMON",  
        name_en="dim_test_dimension",  
        id="0"  
    )  
    response = client.create_design_dimension(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

## Go

根据下列参数，新建维度。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dataartsstudio.NewDataArtsStudioClient(  
        dataartsstudio.DataArtsStudioClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.CreateDesignDimensionRequest{  
        idDatasource:= "0"  
        bizTypeDatasource:= model.GetBizDatasourceRelationVoBizTypeEnum().DIMENSION  
        dbNameDatasource:= "bi"  
        schemaDatasource:= ""  
        datasourcebody := &model.BizDatasourceRelationVo{  
            Id: &idDatasource,  
            BizType: &bizTypeDatasource,  
            DwType: "DWS",  
            DwId: "8259af52bd294f98920ebce75a426391",  
            DbName: &dbNameDatasource,  
            Schema: &schemaDatasource,  
        }  
    }  
    idAttributes:= "0"  
    dimensionIdAttributes:= "0"  
    descriptionAttributes:= ""  
    createByAttributes:= "string"
```

```
isBizPrimaryAttributes:= false
isPartitionKeyAttributes:= false
notNullAttributes:= false
aliasAttributes:= ""
idAttributes1:= "0"
dimensionIdAttributes1:= "0"
descriptionAttributes1:= ""
createByAttributes1:= "string"
isBizPrimaryAttributes1:= false
isPartitionKeyAttributes1:= false
notNullAttributes1:= false
aliasAttributes1:= ""
var listAttributesbody = []model.DimensionAttributeVo{
    {
        Id: &idAttributes,
        DimensionId: &dimensionIdAttributes,
        NameEn: "dim_key",
        NameCh: "测试维度_代理键",
        Description: &descriptionAttributes,
        CreateBy: &createByAttributes,
        DataType: "BIGINT",
        IsPrimaryKey: true,
        IsBizPrimary: &isBizPrimaryAttributes,
        IsPartitionKey: &isPartitionKeyAttributes,
        Ordinal: int32(1),
        NotNull: &notNullAttributes,
        Alias: &aliasAttributes,
    },
    {
        Id: &idAttributes1,
        DimensionId: &dimensionIdAttributes1,
        NameEn: "test_field",
        NameCh: "测试维度_属性",
        Description: &descriptionAttributes1,
        CreateBy: &createByAttributes1,
        DataType: "BIGINT",
        IsPrimaryKey: false,
        IsBizPrimary: &isBizPrimaryAttributes1,
        IsPartitionKey: &isPartitionKeyAttributes1,
        Ordinal: int32(2),
        NotNull: &notNullAttributes1,
        Alias: &aliasAttributes1,
    },
}
configsDimensionVo:= "{}"
aliasDimensionVo:= ""
tableTypeDimensionVo:= "MANAGED"
obsLocationDimensionVo:= ""
ownerDimensionVo:= "test_user"
descriptionDimensionVo:= "无"
idDimensionVo:= "0"
request.Body = &model.DimensionVo{
    Configs: &configsDimensionVo,
    Alias: &aliasDimensionVo,
    TableType: &tableTypeDimensionVo,
    ObsLocation: &obsLocationDimensionVo,
    Owner: &ownerDimensionVo,
    Datasource: datasourcebody,
    Attributes: listAttributesbody,
    L3Id: "1169309252771299328",
    Description: &descriptionDimensionVo,
    NameCh: "测试维度",
    DimensionType: model.GetDimensionVoDimensionTypeEnum().COMMON,
    NameEn: "dim_test_dimension",
    Id: &idDimensionVo,
}
response, err := client.CreateDesignDimension(request)
if err == nil {
    fmt.Printf("%+v\n", response)
```

```
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 DimensionVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.15.3 更新维度

### 功能介绍

根据参数，更新维度信息。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/dimensions

表 8-888 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。



## 请求参数

表 8-889 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-890 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码（新建时只读，更新时必填），ID字符串。
name_en	是	String	维度英文名称。
dimension_type	是	String	维度类型（更新时只能由普通维度改为层级维度，其余场景都不允许修改）。 枚举值： <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
name_ch	是	String	维度名称。
description	是	String	描述。
create_by	否	String	创建人。
update_by	否	String	更新人。

参数	是否必选	参数类型	描述
code_table_id	否	String	引用码表ID（新建/更新“码表维度”时必填），ID字符串。
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。
l3_id	是	String	业务对象ID，ID字符串。
hierarchies	否	Array of <b>DimensionHierarchiesVO</b> objects	层级属性。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_info	否	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	是	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	否	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	是	<b>BizDataSourceRelationVO</b> object	数据源信息，新建/更新时必填。

参数	是否必选	参数类型	描述
owner	是	String	资产责任人，新建/更新时必须填。
obs_location	否	String	外表路径
table_type	否	String	表类型。
distribute	否	String	DISTRIBUTE BY [HASH(column)  REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行 Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	否	String	DISTRIBUTE BY HASH column.
alias	否	String	别名。
self_defined_fields	否	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。
configs	否	String	其他配置
model_id	否	String	所属模型ID
model	否	<a href="#">WorkspaceVO</a> object	模型信息。

表 8-891 DimensionHierarchiesVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name	否	String	层级名称。
attrs	否	Array of <a href="#">HierarchiesAttrVO</a> objects	层级包含的属性。
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-892 HierarchiesAttrVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
hierarchies_id	否	String	层级ID，ID字符串。
attr_id	否	String	属性ID，ID字符串。
level	否	Integer	层次。
attr_name_en	否	String	引用属性编码。
detail_attr_ids	否	Array of strings	详情属性ID，ID字符串。
detail_attr_name_ens	否	Array of strings	详情属性英文。

表 8-893 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>● AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>● ATOMIC_INDEX: 原子指标</li> <li>● ATOMIC_METRIC: 原子指标 (新)</li> <li>● BIZ_CATALOG: 流程架构目录</li> <li>● BIZ_METRIC: 业务指标</li> <li>● CODE_TABLE: 码表</li> <li>● COMMON_CONDITION: 通用限定</li> <li>● COMPOSITE_METRIC: 复合指标 (新)</li> <li>● COMPOUND_METRIC: 复合指标</li> <li>● CONDITION_GROUP: 限定分组</li> <li>● DEGENERATE_DIMENSION: 退化维度</li> <li>● DERIVATIVE_INDEX: 衍生指标</li> <li>● DERIVED_METRIC: 衍生指标 (新)</li> <li>● DIMENSION: 维度</li> <li>● DIMENSION_ATTRIBUTE: 维度属性</li> <li>● DIMENSION_HIERARCHIES: 维度层级</li> <li>● DIMENSION_LOGIC_TABLE: 维度表</li> <li>● DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>● DIRECTORY: 目录</li> <li>● FACT_ATTRIBUTE: 事实表属性</li> <li>● FACT_DIMENSION: 事实表维度</li> <li>● FACT_LOGIC_TABLE: 事实表</li> <li>● FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-894 DimensionAttributeVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
code_table_fiel d_id	否	String	码表属性ID，ID字符串。
name_en	是	String	字段名。
name_ch	是	String	业务属性。
description	否	String	描述。
create_by	否	String	创建人。
data_type	是	String	字段类型。
data_type_ext end	否	String	数据类型扩展字段。
is_primary_ke y	是	Boolean	是否主键。
is_biz_primary	否	Boolean	是否业务主键。
is_partition_k ey	否	Boolean	是否分区。
ordinal	是	Integer	序号。
not_null	否	Boolean	是否不为空。
stand_row_id	否	String	关联的数据标准的ID，ID字符串。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
alias	否	String	别名
self_defined_fi elds	否	Array of <b>SelfDefinedF ieldVO</b> objects	自定义项。



表 8-895 TableMappingVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name	是	String	名称。
description	否	String	描述。
target_table_id	否	String	目的表ID，ID字符串。
src_model_id	否	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	否	String	来源模型名称。
view_text	否	String	采集的视图来源，dws视图逆向使用。
target_table_name	否	String	目的表名称。
details	否	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	否	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	否	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	否	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	否	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-896 TableMappingDetailVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
mapping_id	否	String	名称。
target_attr_id	否	String	目的字段ID，ID字符串。
target_attr_name	是	String	目的字段排序。
src_table_ids	否	String	源表ID。
src_attr_ids	否	String	源表字段ID。
remark	否	String	备注。
create_time	否	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	否	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	否	String	创建人。
update_by	否	String	更新人。
target_attr	否	Object	目标属性。

表 8-897 MappingSourceTableVO

参数	是否必选	参数类型	描述
table1_id	是	String	表1ID，ID字符串。
table2_id	否	String	表2ID，ID字符串。
table1_name	是	String	表1名称。
table2_name	否	String	表2名称。

参数	是否必选	参数类型	描述
join_type	是	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	是	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-898 MappingJoinFieldVO

参数	是否必选	参数类型	描述
field1_id	是	String	属性1ID, ID字符串。
field2_id	是	String	属性2ID, ID字符串。
field1_name	是	String	属性1名称。
field2_name	是	String	属性2名称。

表 8-899 MappingSourceFieldVO

参数	是否必选	参数类型	描述
target_field_id	否	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	是	String	目标字段编码。
field_ids	否	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	否	String	转换表达式。
field_names	否	Array of strings	来源字段名称列表。
changed	否	Boolean	字段是否发生变化。

表 8-900 BizDatasourceRelationVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
biz_id	否	String	业务对象信息，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	是	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	是	String	数据连接ID。
db_name	否	String	数据库名。
queue_name	否	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	否	String	DWS类型需要。

表 8-901 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

表 8-902 WorkspaceVO

参数	是否必选	参数类型	描述
id	否	String	编号，ID字符串。
name	是	String	工作区名字。
description	否	String	描述。
is_physical	否	Boolean	是否为物理表。
frequent	否	Boolean	是否为常用。
top	否	Boolean	分层治理。
level	否	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	否	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_by	否	String	创建人。
update_by	否	String	更新人。
type	是	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	否	String	关联的业务分层的ID列表 { "l1Ids": [], "l2Ids": [], "l3Ids": [] }。

参数	是否必选	参数类型	描述
databases	否	Array of strings	数据库名称数组。
table_model_prefix	否	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

## 响应参数

状态码：200

表 8-903 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-904 data

参数	参数类型	描述
value	<b>DimensionVO</b> object	value，统一的返回结果的外层数据结构。

表 8-905 DimensionVO

参数	参数类型	描述
id	String	编码（新建时只读，更新时必填），ID 字符串。
name_en	String	维度英文名称。
dimension_type	String	维度类型（更新时只能由普通维度改为层级维度，其余场景都不允许修改）。 枚举值： <ul style="list-style-type: none"> <li>COMMON: 普通维度</li> <li>LOOKUP: 码表维度</li> <li>HIERARCHIES: 层级维度</li> </ul>
name_ch	String	维度名称。
description	String	描述。
create_by	String	创建人。



参数	参数类型	描述
update_by	String	更新人。
code_table_id	String	引用码表ID（新建/更新“码表维度”时必填），ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表，只读。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
hierarchies	Array of <b>DimensionHierarchiesVO</b> objects	层级属性。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
l3	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	<b>BizDatasourceRelationVO</b> object	数据源信息，新建/更新时必填。
owner	String	资产责任人，新建/更新时必填。
obs_location	String	外表路径
table_type	String	表类型。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。

参数	参数类型	描述
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	String	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-906 CodeTableVO

参数	参数类型	描述
id	String	码表ID, ID字符串。
name_en	String	表名称, 英文名。
name_ch	String	表名称, 中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>DRAFT: 草稿</li> <li>PUBLISH_DEVELOPING: 发布待审核</li> <li>PUBLISHED: 已发布</li> <li>OFFLINE_DEVELOPING: 下线待审核</li> <li>OFFLINE: 已下线</li> <li>REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-907 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <b>CodeTableFieldValueVO</b> objects	码表属性值。

参数	参数类型	描述
count_field_values	Integer	码表属性值总数。

表 8-908 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-909 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码, ID字符串。
name	String	层级名称。
attrs	Array of <a href="#">HierarchiesAttrVO</a> objects	层级包含的属性。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-910 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码, ID字符串。
hierarchies_id	String	层级ID, ID字符串。

参数	参数类型	描述
attr_id	String	属性ID, ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称, 只读。
detail_attr_ids	Array of strings	详情属性ID, ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文, 只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段, 只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段, 只读。

表 8-911 DimensionAttributeVO

参数	参数类型	描述
id	String	编码, ID字符串。
dimension_id	String	维度ID, 只读, ID字符串。
code_table_field_id	String	码表属性ID, ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>

参数	参数类型	描述
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID, ID字符串。
stand_row_name	String	关联的数据标准名称, 只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息, 只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。

表 8-912 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-913 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-914 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。

表 8-915 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-916 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-917 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-918 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>LEFT: 左外连接</li> <li>RIGHT: 右外连接</li> <li>INNER: 内连接</li> <li>FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-919 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-920 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-921 BizDatasourceRelationVO

参数	参数类型	描述
id	String	编码, ID字符串。

参数	参数类型	描述
biz_id	String	业务对象信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	String	数据连接ID。
dw_name	String	数据连接名，只读。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。

表 8-922 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-923 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

**状态码：400**

**表 8-924** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-925** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-926** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据下列参数，更新维度。

```
PUT https://{endpoint}/v2/{project_id}/design/dimensions
```

```
{
  "id": "1300473264673533952",
  "model_id": "1255482244563357696",
  "name_ch": "测试维度",
  "name_en": "dim_test_dimension",
  "alias": "",
  "owner": "test_user",
  "dimension_type": "COMMON",
```



```

"description": "无",
"l3_id": "1169309252771299328",
"attributes": [ {
  "id": "1300473264694505472",
  "name_ch": "测试维度_代理键",
  "name_en": "dim_key",
  "alias": "",
  "description": "",
  "data_type": "BIGINT",
  "data_type_extend": null,
  "is_biz_primary": false,
  "is_primary_key": true,
  "is_partition_key": false,
  "not_null": false,
  "ordinal": 1,
  "dimension_id": "1300473264673533952",
  "create_by": "string",
  "stand_row_id": null,
  "stand_row_name": null,
  "secrecy_levels": null,
  "self_defined_fields": [ ]
}, {
  "id": "1300473264694505473",
  "name_ch": "测试维度_属性",
  "name_en": "test_field",
  "alias": "",
  "description": "",
  "data_type": "BIGINT",
  "data_type_extend": null,
  "is_biz_primary": false,
  "is_primary_key": false,
  "is_partition_key": false,
  "not_null": false,
  "ordinal": 2,
  "dimension_id": "1300473264673533952",
  "create_by": "string",
  "stand_row_id": null,
  "stand_row_name": null,
  "secrecy_levels": null,
  "self_defined_fields": [ ]
} ],
"datasource": {
  "id": 0,
  "biz_type": "DIMENSION",
  "dw_type": "DWS",
  "dw_id": "8259af52bd294f98920ebce75a426391",
  "db_name": "bi",
  "schema": ""
},
"obs_location": "",
"self_defined_fields": null,
"configs": "{}",
"table_type": "MANAGED",
"mappings": [ ]
}

```

## 响应示例

**状态码: 400**

BadRequest

```

{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}

```

**状态码: 401**

## Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码: 403**

## Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据下列参数，更新维度。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateDesignDimensionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateDesignDimensionRequest request = new UpdateDesignDimensionRequest();
        DimensionVO body = new DimensionVO();
        BizDatasourceRelationVO datasourcebody = new BizDatasourceRelationVO();
        datasourcebody.withId("0")
            .withBizType(BizDatasourceRelationVO.BizTypeEnum.fromValue("DIMENSION"))
            .withDwType("DWS")
            .withDwId("8259af52bd294f98920ebce75a426391")
            .withDbName("bi");
```

```
.withSchema("");
List<DimensionAttributeVO> listbodyAttributes = new ArrayList<>();
listbodyAttributes.add(
    new DimensionAttributeVO()
        .withId("1300473264694505472")
        .withDimensionId("1300473264673533952")
        .withNameEn("dim_key")
        .withNameCh("测试维度_代理键")
        .withDescription("")
        .withCreateBy("string")
        .withDataType("BIGINT")
        .withIsPrimaryKey(true)
        .withIsBizPrimary(false)
        .withIsPartitionKey(false)
        .withOrdinal(1)
        .withNotNull(false)
        .withAlias("")
        .withSelfDefinedFields()
);
listbodyAttributes.add(
    new DimensionAttributeVO()
        .withId("1300473264694505473")
        .withDimensionId("1300473264673533952")
        .withNameEn("test_field")
        .withNameCh("测试维度_属性")
        .withDescription("")
        .withCreateBy("string")
        .withDataType("BIGINT")
        .withIsPrimaryKey(false)
        .withIsBizPrimary(false)
        .withIsPartitionKey(false)
        .withOrdinal(2)
        .withNotNull(false)
        .withAlias("")
        .withSelfDefinedFields()
);
body.withConfigs("{}");
body.withAlias("");
body.withTableType("MANAGED");
body.withObsLocation("");
body.withOwner("test_user");
body.withDatasource(datasourcebody);
body.withAttributes(listbodyAttributes);
body.withL3Id("1169309252771299328");
body.withDescription("无");
body.withNameCh("测试维度");
body.withDimensionType(DimensionVO.DimensionTypeEnum.fromValue("COMMON"));
body.withNameEn("dim_test_dimension");
body.withId("1300473264673533952");
request.withBody(body);
try {
    UpdateDesignDimensionResponse response = client.updateDesignDimension(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据下列参数，更新维度。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateDesignDimensionRequest()
        datasourcebody = BizDatasourceRelationVO(
            id="0",
            biz_type="DIMENSION",
            dw_type="DWS",
            dw_id="8259af52bd294f98920ebce75a426391",
            db_name="bi",
            schema=""
        )
        listAttributesbody = [
            DimensionAttributeVO(
                id="1300473264694505472",
                dimension_id="1300473264673533952",
                name_en="dim_key",
                name_ch="测试维度_代理键",
                description="",
                create_by="string",
                data_type="BIGINT",
                is_primary_key=True,
                is_biz_primary=False,
                is_partition_key=False,
                ordinal=1,
                not_null=False,
                alias="",
            ),
            DimensionAttributeVO(
                id="1300473264694505473",
                dimension_id="1300473264673533952",
                name_en="test_field",
                name_ch="测试维度_属性",
                description="",
                create_by="string",
                data_type="BIGINT",
                is_primary_key=False,
                is_biz_primary=False,
                is_partition_key=False,
                ordinal=2,
                not_null=False,
                alias=""
            )
        ]
```

```
)
]
request.body = DimensionVO(
  configs="{ }",
  alias="",
  table_type="MANAGED",
  obs_location="",
  owner="test_user",
  datasource=datasourcebody,
  attributes=listAttributesbody,
  l3_id="1169309252771299328",
  description="无",
  name_ch="测试维度",
  dimension_type="COMMON",
  name_en="dim_test_dimension",
  id="1300473264673533952"
)
response = client.update_design_dimension(request)
print(response)
except exceptions.ClientRequestException as e:
  print(e.status_code)
  print(e.request_id)
  print(e.error_code)
  print(e.error_msg)
```

## Go

根据下列参数，更新维度。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDesignDimensionRequest{
        idDatasource:= "0"
        bizTypeDatasource:= model.GetBizDatasourceRelationVoBizTypeEnum().DIMENSION
        dbNameDatasource:= "bi"
        schemaDatasource:= ""
        datasourcebody := &model.BizDatasourceRelationVo{
            Id: &idDatasource,
            BizType: &bizTypeDatasource,
```

```

DwType: "DWS",
Dwld: "8259af52bd294f98920ebce75a426391",
DbName: &dbNameDatasource,
Schema: &schemaDatasource,
}
idAttributes:= "1300473264694505472"
dimensionIdAttributes:= "1300473264673533952"
descriptionAttributes:= ""
createByAttributes:= "string"
isBizPrimaryAttributes:= false
isPartitionKeyAttributes:= false
notNullAttributes:= false
aliasAttributes:= ""
idAttributes1:= "1300473264694505473"
dimensionIdAttributes1:= "1300473264673533952"
descriptionAttributes1:= ""
createByAttributes1:= "string"
isBizPrimaryAttributes1:= false
isPartitionKeyAttributes1:= false
notNullAttributes1:= false
aliasAttributes1:= ""
var listAttributesbody = []model.DimensionAttributeVo{
{
  Id: &idAttributes,
  DimensionId: &dimensionIdAttributes,
  NameEn: "dim_key",
  NameCh: "测试维度_代理键",
  Description: &descriptionAttributes,
  CreateBy: &createByAttributes,
  DataType: "BIGINT",
  IsPrimaryKey: true,
  IsBizPrimary: &isBizPrimaryAttributes,
  IsPartitionKey: &isPartitionKeyAttributes,
  Ordinal: int32(1),
  NotNull: &notNullAttributes,
  Alias: &aliasAttributes,
},
{
  Id: &idAttributes1,
  DimensionId: &dimensionIdAttributes1,
  NameEn: "test_field",
  NameCh: "测试维度_属性",
  Description: &descriptionAttributes1,
  CreateBy: &createByAttributes1,
  DataType: "BIGINT",
  IsPrimaryKey: false,
  IsBizPrimary: &isBizPrimaryAttributes1,
  IsPartitionKey: &isPartitionKeyAttributes1,
  Ordinal: int32(2),
  NotNull: &notNullAttributes1,
  Alias: &aliasAttributes1,
},
}
configsDimensionVo:= "{}"
aliasDimensionVo:= ""
tableTypeDimensionVo:= "MANAGED"
obsLocationDimensionVo:= ""
ownerDimensionVo:= "test_user"
descriptionDimensionVo:= "无"
idDimensionVo:= "1300473264673533952"
request.Body = &model.DimensionVo{
  Configs: &configsDimensionVo,
  Alias: &aliasDimensionVo,
  TableType: &tableTypeDimensionVo,
  ObsLocation: &obsLocationDimensionVo,
  Owner: &ownerDimensionVo,
  Datasource: datasourcebody,
  Attributes: listAttributesbody,
  L3Id: "1169309252771299328",
}

```

```
Description: &descriptionDimensionVo,  
NameCh: "测试维度",  
DimensionType: model.GetDimensionVoDimensionTypeEnum().COMMON,  
NameEn: "dim_test_dimension",  
Id: &idDimensionVo,  
}  
response, err := client.UpdateDesignDimension(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 DimensionVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.15.4 删除维度

### 功能介绍

根据传入的维度ID，删除维度。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/dimensions

表 8-927 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-928 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-929 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200



表 8-930 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-931 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

**状态码：400**

表 8-932 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-933 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-934 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入的参数，删除维度，只支持删除状态为草稿、已下线、已驳回的维度。

```
DELETE https://{endpoint}/v2/{project_id}/design/dimensions
{
  "ids" : [ "1227990857618227200" ]
}
```

## 响应示例

### 状态码：200

Success，返回data数据是删除个数。

```
{
  "value" : 1
}
```

### 状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据传入的参数，删除维度，只支持删除状态为草稿、已下线、已驳回的维度。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteDesignDimensionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteDesignDimensionRequest request = new DeleteDesignDimensionRequest();
        IdsParam body = new IdsParam();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("1227990857618227200");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            DeleteDesignDimensionResponse response = client.deleteDesignDimension(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据传入的参数，删除维度，只支持删除状态为草稿、已下线、已驳回的维度。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteDesignDimensionRequest()
    listIdsbody = [
        "1227990857618227200"
    ]
    request.body = IdsParam(
        ids=listIdsbody
    )
    response = client.delete_design_dimension(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据传入的参数，删除维度，只支持删除状态为草稿、已下线、已驳回的维度。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDesignDimensionRequest{}
    var listIdsbody = []string{
        "1227990857618227200",
    }
    request.Body = &model.IdsParam{
        Ids: listIdsbody,
    }
}
```

```
}  
response, err := client.DeleteDesignDimension(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.15.5 查看维度详情

### 功能介绍

通过ID查看维度详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/dimensions/{id}

表 8-935 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-936 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

## 请求参数

表 8-937 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-938 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-939 data

参数	参数类型	描述
value	<b>DimensionVO</b> object	value, 统一的返回结果的外层数据结构。

表 8-940 DimensionVO

参数	参数类型	描述
id	String	编码 (新建时只读, 更新时必填), ID 字符串。
name_en	String	维度英文名称。
dimension_type	String	维度类型 (更新时只能由普通维度改为层级维度, 其余场景都不允许修改)。 枚举值: <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
name_ch	String	维度名称。
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
code_table_id	String	引用码表ID (新建/更新“码表维度”时必填), ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表, 只读。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象ID, ID字符串。
hierarchies	Array of <b>DimensionHierar chiesVO</b> objects	层级属性。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	<b>BizDataSourceRelationVO</b> object	数据源信息，新建/更新时必填。
owner	String	资产责任人，新建/更新时必填。
obs_location	String	外表路径
table_type	String	表类型。



参数	参数类型	描述
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	String	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-941 CodeTableVO

参数	参数类型	描述
id	String	码表ID，ID字符串。
name_en	String	表名称，英文名。
name_ch	String	表名称，中文名。

参数	参数类型	描述
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-942 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必须填), ID字符串。

参数	参数类型	描述
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-943 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-944 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码, ID字符串。
name	String	层级名称。

参数	参数类型	描述
attrs	Array of <b>HierarchiesAttrVO</b> objects	层级包含的属性。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-945 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码，ID字符串。
hierarchies_id	String	层级ID，ID字符串。
attr_id	String	属性ID，ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称，只读。
detail_attr_ids	Array of strings	详情属性ID，ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文，只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段，只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段，只读。

表 8-946 DimensionAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
dimension_id	String	维度ID，只读，ID字符串。
code_table_field_id	String	码表属性ID，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。

表 8-947 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。



参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-948 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-949 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。

表 8-950 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-951 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-952 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-953 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。



参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>LEFT: 左外连接</li> <li>RIGHT: 右外连接</li> <li>INNER: 内连接</li> <li>FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-954 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-955 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-956 BizDatasourceRelationVO

参数	参数类型	描述
id	String	编码, ID字符串。

参数	参数类型	描述
biz_id	String	业务对象信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	String	数据连接ID。
dw_name	String	数据连接名，只读。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。

表 8-957 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-958 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

### 状态码：400

表 8-959 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

### 状态码：401

表 8-960 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

### 状态码：403

表 8-961 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据维度ID，查询维度信息。

```
GET https://{endpoint}/v2/{project_id}/design/dimensions/1222930977286987776
```

## 响应示例

### 状态码：200

Success，返回data数据是DimensionVO详情。

```
{  
  "data" : {
```

```

"value": {
  "id": "1222930977286987776",
  "name_en": "dim_test_czh_0328",
  "dimension_type": "COMMON",
  "name_ch": "test_czh_0328",
  "description": "无",
  "create_by": "test_uesr",
  "update_by": null,
  "code_table_id": null,
  "code_table": null,
  "l1_id": "1194966710038474752",
  "l2_id": null,
  "l3_id": null,
  "hierarchies": [ ],
  "status": "PUBLISHED",
  "create_time": "2024-03-28T15:31:06+08:00",
  "update_time": "2024-03-28T15:31:06+08:00",
  "approval_info": {
    "id": "1222930978226511872",
    "tenant_id": null,
    "name_ch": null,
    "name_en": null,
    "biz_id": "1222930977286987776",
    "biz_type": null,
    "biz_info": null,
    "biz_info_obj": null,
    "biz_version": 0,
    "biz_status": "PUBLISHED",
    "approval_status": "APPROVED",
    "approval_type": null,
    "submit_time": "2024-03-28T15:31:06+08:00",
    "create_by": null,
    "l1": null,
    "l2": null,
    "l3": null,
    "approval_time": "2024-03-28T15:31:06+08:00",
    "approver": "test_uesr",
    "email": null,
    "msg": "Fast Approval OK.",
    "directory_path": null
  },
  "new_biz": null,
  "l1": "test_czh_catalog",
  "l2": null,
  "l3": null,
  "attributes": [ {
    "id": "1222930977370873856",
    "dimension_id": "1222930977286987776",
    "code_table_field_id": null,
    "name_en": "dim_test_czh_0328_key",
    "name_ch": "test_czh_0328",
    "description": "",
    "create_by": "test_uesr",
    "data_type": "BIGINT",
    "domain_type": "NUMBER",
    "data_type_extend": null,
    "is_primary_key": true,
    "is_biz_primary": false,
    "is_partition_key": false,
    "ordinal": 1,
    "not_null": false,
    "stand_row_id": null,
    "stand_row_name": null,
    "quality_infos": null,
    "secrecy_levels": [ ],
    "status": null,
    "create_time": "2024-03-28T15:31:06+08:00",
    "update_time": "2024-03-28T15:31:06+08:00",
    "alias": ""
  }

```

```

"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : ""
}
]
}, {
  "id" : "1222930977370873857",
  "dimension_id" : "1222930977286987776",
  "code_table_field_id" : null,
  "name_en" : "aaa",
  "name_ch" : "aaa",
  "description" : "",
  "create_by" : "test_uesr",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : null,
  "is_primary_key" : false,
  "is_biz_primary" : false,
  "is_partition_key" : false,
  "ordinal" : 2,
  "not_null" : false,
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "secrecy_levels" : [ ],
  "status" : null,
  "create_time" : "2024-03-28T15:31:06+08:00",
  "update_time" : "2024-03-28T15:31:06+08:00",
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "aa",
    "fd_name_en" : "aa",
    "not_null" : false,
    "fd_value" : ""
  }, {
    "fd_name_ch" : "bb",
    "fd_name_en" : "bb",
    "not_null" : false,
    "fd_value" : ""
  }
]
}, {
  "id" : "1222930977370873858",
  "dimension_id" : "1222930977286987776",
  "code_table_field_id" : null,
  "name_en" : "bbb",
  "name_ch" : "bbb",
  "description" : "",
  "create_by" : "test_uesr",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : null,
  "is_primary_key" : false,
  "is_biz_primary" : false,
  "is_partition_key" : false,
  "ordinal" : 3,
  "not_null" : false,
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "secrecy_levels" : [ ],
  "status" : null,
  "create_time" : "2024-03-28T15:31:06+08:00",
  "update_time" : "2024-03-28T15:31:06+08:00",

```



```

"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : ""
}
]
}],
"mappings" : [ ],
"datasource" : {
  "id" : "1222930977387651072",
  "biz_id" : "1222930977286987776",
  "biz_type" : "DIMENSION",
  "dw_type" : "DWS",
  "dw_id" : "8259af52bd294f98920ebce75a426391",
  "dw_name" : "dws",
  "db_name" : "bi",
  "queue_name" : "default",
  "schema" : ""
},
"owner" : "afcaca",
"table_type" : "MANAGED",
"distribute" : null,
"distribute_column" : null,
"obs_location" : "",
"alias" : "",
"configs" : "{}",
"self_defined_fields" : [ {
  "fd_name_ch" : "自定义项1",
  "fd_name_en" : "selfDefine1",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "自定义项2",
  "fd_name_en" : "selfDefine2",
  "not_null" : true,
  "fd_value" : "adada"
}
],
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
}
}
}

```

### 状态码：400

#### BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

### 状态码：401

#### Unauthorized

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}

```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**SDK 代码示例**

SDK代码示例如下。

**Java**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowDimensionByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDimensionByIdRequest request = new ShowDimensionByIdRequest();
        request.withId("{id}");
        try {
            ShowDimensionByIdResponse response = client.showDimensionById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowDimensionByIdRequest()
        request.id = "{id}"
        response = client.show_dimension_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```

        WithCredential(auth).
        Build()

        request := &model.ShowDimensionByIdRequest{}
        request.Id = "{id}"
        response, err := client.ShowDimensionById(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是DimensionVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.15.6 查看维度颗粒度

### 功能介绍

查询维度颗粒度，依据tableId查询涉及所有维度，不传tableId查询所有维度组颗粒度。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/dimension/groups

表 8-962 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-963 Query 参数

参数	是否必选	参数类型	描述
table_id	否	String	关联表的ID, ID字符串。
biz_type	否	String	按业务类型查询, 可选业务类型有: ATOMIC_INDEX (原子指标)、DERIVATIVE_INDEX (衍生指标)、DIMENSION (维度)、TIME_CONDITION (时间限定)、DIMENSION_LOGIC_TABLE (维度表)、FACT_LOGIC_TABLE (事实表)、AGGREGATION_LOGIC_TABLE (汇总表)、TABLE_MODEL (关系建模表)、CODE_TABLE (码表)、STANDARD_ELEMENT (数据标准)、BIZ_METRIC (业务指标)、COMPOUND_METRIC (复合指标)、SUBJECT (主题)、ATOMIC_METRIC (原子指标(新))、DERIVED_METRIC (衍生指标(新))、COMPOSITE_METRIC (复合指标(新))。
limit	否	Integer	每页查询条数, 即查询Y条数据。默认值50, 取值范围[1,100]。
offset	否	Integer	查询起始坐标, 即跳过X条数据, 仅支持0或limit的整数倍, 不满足则向下取整, 默认值0。

## 请求参数

表 8-964 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 (响应消息头中X-Subject-Token的值) 使用Token认证时必选。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-965 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-966 data

参数	参数类型	描述
value	<b>value</b> object	value，统一的返回结果的外层数据结构。

表 8-967 value

参数	参数类型	描述
main_table	<b>FactLogicTableVO</b> object	事实表。
dimension_tables	Array of <b>DimensionLogicTableVO</b> objects	维度。

表 8-968 FactLogicTableVO

参数	参数类型	描述
id	String	事实表ID, ID字符串。
tb_name	String	表名称。
tb_logic_name	String	逻辑实体名。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象guid, ID字符串。
description	String	描述。
dw_name	String	数据连接名称, 只读, 创建和更新时无需填写。
owner	String	资产责任人。
create_by	String	创建人。
queue_name	String	dli数据连接执行sql所需的队列, 数据连接类型为DLI时必须。
dw_id	String	数据连接ID。
db_name	String	库名。
tb_id	String	数据表ID, 只读。
dim_table_ids	Array of strings	关联维度表ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
schema	String	DWS类型需要。

参数	参数类型	描述
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
table_type	String	表类型。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>DRAFT: 草稿</li> <li>PUBLISH_DEVELOPING: 发布待审核</li> <li>PUBLISHED: 已发布</li> <li>OFFLINE_DEVELOPING: 下线待审核</li> <li>OFFLINE: 已下线</li> <li>REJECT: 已驳回</li> </ul>
tb_guid	String	表发布后，创建的数据目录技术资产guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产guid，只读，创建和更新时无需填写。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理, 只读。
dimensions	Array of <b>FactTableAttribut eVO</b> objects	事实表维度信息, 只读。
attributes	Array of <b>FactTableAttribut eVO</b> objects	事实表事实属性信息, 只读。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
measures	Array of <b>FactTableAttribut eVO</b> objects	事实表度量信息, 只读。
table_attributes	Array of <b>FactTableAttribut eVO</b> objects	事实表所有属性。
configs	String	其他配置
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
quality_id	String	质量ID, ID字符串。
alias	String	别名。

参数	参数类型	描述
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
obs_location	String	外表路径
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
model_id	Long	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-969 FactTableAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
fact_logic_table_id	String	所属事实表ID，只读，ID字符串。
ordinal	Integer	序号。
dimension_id	String	维度ID，ID字符串。
dimension_attr_id	String	维度属性ID，ID字符串。
role	String	维度角色。
dimension	<b>DimensionVO</b> object	dimension_id对应的维度信息，只读。
is_primary_key	Boolean	是否主键。

参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键，只读。
secrecy_levels	Array of <a href="#">SecrecyLevelRecordVO</a> objects	密级
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
name_en	String	英文名。
name_ch	String	中文名。
not_null	Boolean	是否不为空。

参数	参数类型	描述
attribute_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。

表 8-970 DimensionVO

参数	参数类型	描述
id	String	编码（新建时只读，更新时必填），ID字符串。
name_en	String	维度英文名称。
dimension_type	String	维度类型（更新时只能由普通维度改为层级维度，其余场景都不允许修改）。 枚举值： <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
name_ch	String	维度名称。

参数	参数类型	描述
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
code_table_id	String	引用码表ID（新建/更新“码表维度”时必须填），ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表，只读。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
hierarchies	Array of <b>DimensionHierar chiesVO</b> objects	层级属性。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。



参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	<b>BizDatasourceRelationVO</b> object	数据源信息，新建/更新时必填。
owner	String	资产责任人，新建/更新时必填。
obs_location	String	外表路径
table_type	String	表类型。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。

参数	参数类型	描述
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	String	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-971 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	层级名称。
attrs	Array of <b>HierarchiesAttrVO</b> objects	层级包含的属性。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-972 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码，ID字符串。
hierarchies_id	String	层级ID，ID字符串。
attr_id	String	属性ID，ID字符串。
level	Integer	层次。

参数	参数类型	描述
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称，只读。
detail_attr_ids	Array of strings	详情属性ID，ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文，只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段，只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段，只读。

表 8-973 DimensionAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
dimension_id	String	维度ID，只读，ID字符串。
code_table_field_id	String	码表属性ID，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。

参数	参数类型	描述
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。

表 8-974 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-975 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-976 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>LEFT: 左外连接</li> <li>RIGHT: 右外连接</li> <li>INNER: 内连接</li> <li>FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-977 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-978 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-979 BizDatasourceRelationVO

参数	参数类型	描述
id	String	编码, ID字符串。

参数	参数类型	描述
biz_id	String	业务对象信息，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	String	数据连接ID。
dw_name	String	数据连接名，只读。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。

表 8-980 SecrecyLevelRecordVO

参数	参数类型	描述
id	String	属性关联密级的ID，ID字符串。
secrecyLevel_id	String	密级的ID，ID字符串。
secrecyLevel_name	String	密级名称。
uuid	String	数据安全主键。
slevel	Integer	密级等级。
description	String	密级描述。

参数	参数类型	描述
biz_id	String	业务对象ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建者。
update_by	String	更新者。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-981 DimensionLogicTableVO

参数	参数类型	描述
id	String	表ID
tb_name	String	表名称。
tb_logic_name	String	逻辑实体名。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
create_by	String	创建人。
description	String	描述。

参数	参数类型	描述
dimension_id	String	所属维度ID，ID字符串。
owner	String	资产责任人。
dimension_type	String	维度类型。 枚举值： <ul style="list-style-type: none"> <li>COMMON: 普通维度</li> <li>LOOKUP: 码表维度</li> <li>HIERARCHIES: 层级维度</li> </ul>
code_table_id	String	引用码表ID（新建/更新“码表维度”时必须填），ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表，只读。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
dw_id	String	数据连接ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
db_name	String	库名。
tb_id	String	数据表ID，只读。
schema	String	DWS类型需要。
table_type	String	表类型。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>DRAFT: 草稿</li> <li>PUBLISH_DEVELOPING: 发布待审核</li> <li>PUBLISHED: 已发布</li> <li>OFFLINE_DEVELOPING: 下线待审核</li> <li>OFFLINE: 已下线</li> <li>REJECT: 已驳回</li> </ul>

参数	参数类型	描述
tb_guid	String	表发布后，创建的数据目录技术资产 guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产 guid，只读，创建和更新时无需填写。
dimension_name	String	关联维度名称，只读。
attributes	Array of <b>DimensionLogicTableAttributeVO</b> objects	字段属性。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

参数	参数类型	描述
physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
quality_id	String	质量ID，ID字符串。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
obs_location	String	外表路径

参数	参数类型	描述
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>• INVALID_TYPE: 无效环境</li> <li>• DEV_TYPE: 开发环境</li> <li>• PROD_TYPE: 生产环境</li> <li>• DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	Long	所属模型ID。
model	<a href="#">WorkspaceVO</a> object	模型信息。

表 8-982 CodeTableVO

参数	参数类型	描述
id	String	码表ID，ID字符串。
name_en	String	表名称，英文名。
name_ch	String	表名称，中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID，ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-983 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID，ID字符串。
code_table_id	String	所属码表ID（更新时必须填），ID字符串。
ordinal	Integer	序号。
name_en	String	字段名，英文。
name_ch	String	字段名，中文。
description	String	描述。
data_type	String	字段类型。

参数	参数类型	描述
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-984 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-985 DimensionLogicTableAttributeVO

参数	参数类型	描述
id	String	维度表ID, ID字符串。
dimension_logic_table_id	String	所属维表ID。
ordinal	Integer	序号
dimension_attribute_id	String	维度属性ID, ID字符串。
name_en	String	字段名, 只读。
name_ch	String	业务属性, 只读。

参数	参数类型	描述
description	String	描述，只读。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键，只读。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否主键分区，只读。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级

表 8-986 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-987 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理, 只读。

表 8-988 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数 “project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-989 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-990 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-991 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-992 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-993 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-994 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。



参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

查询表ID为1169329693619625984、业务类型为FACT\_LOGIC\_TABLE的维度颗粒度。

```
GET https://{endpoint}/v2/{project_id}/design/dimension/groups?
biz_type=FACT_LOGIC_TABLE&table_id=1169329693619625984
```

## 响应示例

**状态码：200**

Success, 返回data数据是DerivativeIndexDimensionVO详情。

```
{
  "data": {
    "value": {
      "main_table": {
        "id": "1169329693619625984",
        "tb_name": "fact_stroke_order",
        "tb_logic_name": "行程订单",
        "l1_id": null,
        "l2_id": null,
        "l3_id": "1179816339158790144",
        "description": "无",
        "dw_name": null,
        "owner": "test_uesr",
        "create_by": "test_uesr",
        "queue_name": "",
        "dw_id": "d800f74aedca4e68a95ecbd0569cabfa",
        "db_name": "default",
        "tb_id": "NativeTable-d800f74aedca4e68a95ecbd0569cabfa-default-fact_stroke_order",
        "dim_table_ids": null,
        "reversed": null,
        "partition_conf": null,
        "dirty_out_switch": false,
        "dirty_out_database": "",
        "dirty_out_prefix": "",
        "dirty_out_suffix": "",
        "schema": "",
        "distribute": null,
        "distribute_column": null,
        "obs_location": null,
        "table_type": "HIVE_TABLE",
        "status": "PUBLISHED",
        "tb_guid": "fe241223-c87c-4011-90a3-ced201bd4fd0",
        "tb_logic_guid": "5231b940-c716-4d7d-bbce-d0b46f5dec5c",
        "dw_type": "MRS_HIVE",
        "l1": null,
        "l2": null,
        "l3": null,
        "create_time": "2023-11-01T17:38:44+08:00",
        "update_time": "2023-11-30T16:13:31+08:00",
        "approval_info": {
          "id": "1179817434769715200",
          "tenant_id": null,
          "name_ch": null,
          "name_en": null,
          "biz_id": "1169329693619625984",
          "biz_type": null,
          "biz_info": null,
        }
      }
    }
  }
}
```

```

    "biz_info_obj" : null,
    "biz_version" : 0,
    "biz_status" : "PUBLISHED",
    "approval_status" : "APPROVED",
    "approval_type" : null,
    "submit_time" : "2023-11-30T16:13:16+08:00",
    "create_by" : null,
    "l1" : null,
    "l2" : null,
    "l3" : null,
    "approval_time" : "2023-11-30T16:13:16+08:00",
    "approver" : "test_uesr",
    "email" : null,
    "msg" : "Fast Approval OK.",
    "directory_path" : null
  },
  "new_biz" : null,
  "dimensions" : [ {
    "id" : "1169329693695123456",
    "fact_logic_table_id" : "1169329693619625984",
    "ordinal" : 1,
    "dimension_id" : "1169323046352371712",
    "role" : null,
    "dimension" : null,
    "is_primary_key" : false,
    "is_partition_key" : false,
    "is_foreign_key" : true,
    "secrecy_levels" : [ ],
    "description" : "",
    "data_type" : "BIGINT",
    "domain_type" : "NUMBER",
    "data_type_extend" : null,
    "name_en" : "rate_code_id",
    "name_ch" : "费率id",
    "not_null" : false,
    "attribute_type" : "FACT_DIMENSION",
    "stand_row_id" : null,
    "stand_row_name" : null,
    "quality_infos" : null,
    "alias" : "",
    "self_defined_fields" : [ ]
  }, {
    "id" : "1169329693695123457",
    "fact_logic_table_id" : "1169329693619625984",
    "ordinal" : 2,
    "dimension_id" : "1169322847009742848",
    "role" : null,
    "dimension" : null,
    "is_primary_key" : false,
    "is_partition_key" : false,
    "is_foreign_key" : true,
    "secrecy_levels" : [ ],
    "description" : "",
    "data_type" : "BIGINT",
    "domain_type" : "NUMBER",
    "data_type_extend" : null,
    "name_en" : "vendor_id",
    "name_ch" : "供应商id",
    "not_null" : false,
    "attribute_type" : "FACT_DIMENSION",
    "stand_row_id" : null,
    "stand_row_name" : null,
    "quality_infos" : null,
    "alias" : "",
    "self_defined_fields" : [ ]
  }, {
    "id" : "1169329693695123458",
    "fact_logic_table_id" : "1169329693619625984",
    "ordinal" : 3,

```

```

"dimension_id" : "1169323245690863616",
"role" : null,
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : true,
"secrecy_levels" : [ ],
"description" : "",
"data_type" : "BIGINT",
"domain_type" : "NUMBER",
"data_type_extend" : null,
"name_en" : "payment_type_id",
"name_ch" : "付款方式编码",
"not_null" : false,
"attribute_type" : "FACT_DIMENSION",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}, {
"id" : "1169329693695123459",
"fact_logic_table_id" : "1169329693619625984",
"ordinal" : 4,
"dimension_id" : "1169327565677182976",
"role" : "dim_pickup",
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : true,
"secrecy_levels" : [ ],
"description" : "日期层维表",
"data_type" : "TIMESTAMP",
"domain_type" : "DATETIME",
"data_type_extend" : null,
"name_en" : "dim_pickup_date_key",
"name_ch" : "上车时间",
"not_null" : false,
"attribute_type" : "FACT_DIMENSION",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}, {
"id" : "1169329693695123460",
"fact_logic_table_id" : "1169329693619625984",
"ordinal" : 5,
"dimension_id" : "1169327565677182976",
"role" : "dim_dropoff",
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : true,
"secrecy_levels" : [ ],
"description" : "日期层维表",
"data_type" : "TIMESTAMP",
"domain_type" : "DATETIME",
"data_type_extend" : null,
"name_en" : "dim_dropoff_date_key",
"name_ch" : "下车时间",
"not_null" : false,
"attribute_type" : "FACT_DIMENSION",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}],

```

```

"attributes" : null,
"mappings" : null,
"measures" : [ {
  "id" : "1169329693695123461",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 6,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : "STRING",
  "data_type_extend" : "",
  "name_en" : "pu_location_id",
  "name_ch" : "上车地点",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123462",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 7,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : "STRING",
  "data_type_extend" : "",
  "name_en" : "do_location_id",
  "name_ch" : "下车地点",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123463",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 8,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "fare_amount",
  "name_ch" : "车费",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",

```

```

"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123464",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 9,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "extra",
  "name_ch" : "加收",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123465",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 10,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "mta_tax",
  "name_ch" : "MTA税",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123466",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 11,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "tip_amount",

```

```

"name_ch": "手续费",
"not_null": false,
"attribute_type": "FACT_MEASURE",
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": null,
"alias": "",
"self_defined_fields": [ ]
}, {
"id": "1169329693695123467",
"fact_logic_table_id": "1169329693619625984",
"ordinal": 12,
"dimension_id": null,
"role": null,
"dimension": null,
"is_primary_key": false,
"is_partition_key": false,
"is_foreign_key": null,
"secrecy_levels": [ ],
"description": "",
"data_type": "DECIMAL",
"domain_type": "NUMBER",
"data_type_extend": "(10,2)",
"name_en": "tolls_amount",
"name_ch": "通行费",
"not_null": false,
"attribute_type": "FACT_MEASURE",
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": null,
"alias": "",
"self_defined_fields": [ ]
}, {
"id": "1169329693695123468",
"fact_logic_table_id": "1169329693619625984",
"ordinal": 13,
"dimension_id": null,
"role": null,
"dimension": null,
"is_primary_key": false,
"is_partition_key": false,
"is_foreign_key": null,
"secrecy_levels": [ ],
"description": "",
"data_type": "DECIMAL",
"domain_type": "NUMBER",
"data_type_extend": "(10,2)",
"name_en": "improvement_surcharge",
"name_ch": "改善附加费",
"not_null": false,
"attribute_type": "FACT_MEASURE",
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": null,
"alias": "",
"self_defined_fields": [ ]
}, {
"id": "1169329693695123469",
"fact_logic_table_id": "1169329693619625984",
"ordinal": 14,
"dimension_id": null,
"role": null,
"dimension": null,
"is_primary_key": false,
"is_partition_key": false,
"is_foreign_key": null,
"secrecy_levels": [ ],
"description": "",
"data_type": "DECIMAL",

```

```

"domain_type": "NUMBER",
"data_type_extend": "(10,2)",
"name_en": "total_amount",
"name_ch": "总车费",
"not_null": false,
"attribute_type": "FACT_MEASURE",
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": null,
"alias": "",
"self_defined_fields": []
}],
"table_attributes": [ {
"id": "1169329693695123456",
"fact_logic_table_id": "1169329693619625984",
"ordinal": 1,
"dimension_id": "1169323046352371712",
"role": null,
"dimension": null,
"is_primary_key": false,
"is_partition_key": false,
"is_foreign_key": true,
"secrecy_levels": [],
"description": "",
"data_type": "BIGINT",
"domain_type": "NUMBER",
"data_type_extend": null,
"name_en": "rate_code_id",
"name_ch": "费率id",
"not_null": false,
"attribute_type": "FACT_DIMENSION",
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": null,
"alias": "",
"self_defined_fields": []
}, {
"id": "1169329693695123457",
"fact_logic_table_id": "1169329693619625984",
"ordinal": 2,
"dimension_id": "1169322847009742848",
"role": null,
"dimension": null,
"is_primary_key": false,
"is_partition_key": false,
"is_foreign_key": true,
"secrecy_levels": [],
"description": "",
"data_type": "BIGINT",
"domain_type": "NUMBER",
"data_type_extend": null,
"name_en": "vendor_id",
"name_ch": "供应商id",
"not_null": false,
"attribute_type": "FACT_DIMENSION",
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": null,
"alias": "",
"self_defined_fields": []
}, {
"id": "1169329693695123458",
"fact_logic_table_id": "1169329693619625984",
"ordinal": 3,
"dimension_id": "1169323245690863616",
"role": null,
"dimension": null,
"is_primary_key": false,
"is_partition_key": false,

```

```

"is_foreign_key" : true,
"secrecy_levels" : [ ],
"description" : "",
"data_type" : "BIGINT",
"domain_type" : "NUMBER",
"data_type_extend" : null,
"name_en" : "payment_type_id",
"name_ch" : "付款方式编码",
"not_null" : false,
"attribute_type" : "FACT_DIMENSION",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123459",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 4,
  "dimension_id" : "1169327565677182976",
  "role" : "dim_pickup",
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : true,
  "secrecy_levels" : [ ],
  "description" : "日期层维表",
  "data_type" : "TIMESTAMP",
  "domain_type" : "DATETIME",
  "data_type_extend" : null,
  "name_en" : "dim_pickup_date_key",
  "name_ch" : "上车时间",
  "not_null" : false,
  "attribute_type" : "FACT_DIMENSION",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123460",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 5,
  "dimension_id" : "1169327565677182976",
  "role" : "dim_dropoff",
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : true,
  "secrecy_levels" : [ ],
  "description" : "日期层维表",
  "data_type" : "TIMESTAMP",
  "domain_type" : "DATETIME",
  "data_type_extend" : null,
  "name_en" : "dim_dropoff_date_key",
  "name_ch" : "下车时间",
  "not_null" : false,
  "attribute_type" : "FACT_DIMENSION",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123461",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 6,
  "dimension_id" : null,
  "role" : null,

```



```
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : null,
"secrecy_levels" : [ ],
"description" : "",
"data_type" : "STRING",
"domain_type" : "STRING",
"data_type_extend" : "",
"name_en" : "pu_location_id",
"name_ch" : "上车地点",
"not_null" : false,
"attribute_type" : "FACT_MEASURE",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123462",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 7,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : "STRING",
  "data_type_extend" : "",
  "name_en" : "do_location_id",
  "name_ch" : "下车地点",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123463",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 8,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "fare_amount",
  "name_ch" : "车费",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123464",
  "fact_logic_table_id" : "1169329693619625984",
```

```
"ordinal" : 9,
"dimension_id" : null,
"role" : null,
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : null,
"secrecy_levels" : [ ],
"description" : "",
"data_type" : "DECIMAL",
"domain_type" : "NUMBER",
"data_type_extend" : "(10,2)",
"name_en" : "extra",
"name_ch" : "加收",
"not_null" : false,
"attribute_type" : "FACT_MEASURE",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123465",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 10,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "mta_tax",
  "name_ch" : "MTA税",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123466",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 11,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "tip_amount",
  "name_ch" : "手续费",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
```

```

}, {
  "id" : "1169329693695123467",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 12,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "tolls_amount",
  "name_ch" : "通行费",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123468",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 13,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "improvement_surcharge",
  "name_ch" : "改善附加费",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ ]
}, {
  "id" : "1169329693695123469",
  "fact_logic_table_id" : "1169329693619625984",
  "ordinal" : 14,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "DECIMAL",
  "domain_type" : "NUMBER",
  "data_type_extend" : "(10,2)",
  "name_en" : "total_amount",
  "name_ch" : "总车费",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,

```

```

    "quality_infos": null,
    "alias": "",
    "self_defined_fields": [ ]
  } ],
  "physical_table": "UPDATE_SUCCESS",
  "dev_physical_table": "NO_NEED",
  "technical_asset": "CREATE_SUCCESS",
  "business_asset": "UPDATE_SUCCESS",
  "meta_data_link": "UPDATE_SUCCESS",
  "data_quality": "NO_NEED",
  "summary_status": "SUMMARY_SUCCESS",
  "quality_id": null,
  "alias": "",
  "configs": "{}",
  "self_defined_fields": [ ],
  "dev_version": null,
  "prod_version": null,
  "dev_version_name": null,
  "prod_version_name": null,
  "env_type": "INVALID_TYPE"
},
"dimension_tables": [ {
  "id": "1169323047912652800",
  "tb_name": "dim_rate_code",
  "tb_logic_name": "费率代码",
  "l1_id": "1169309252771299328",
  "l2_id": "1169309574931652608,1169311054568800256",
  "l3_id": null,
  "create_by": "test_uesr",
  "description": "无",
  "dimension_id": "1169323046352371712",
  "owner": "test_uesr",
  "dimension_type": "LOOKUP",
  "code_table_id": "1169315702553886720",
  "code_table": null,
  "queue_name": "",
  "dw_id": "d800f74aedca4e68a95ecbd0569cabfa",
  "reversed": null,
  "partition_conf": null,
  "dirty_out_switch": false,
  "dirty_out_database": "",
  "dirty_out_prefix": "",
  "dirty_out_suffix": "",
  "db_name": "default",
  "tb_id": "NativeTable-d800f74aedca4e68a95ecbd0569cabfa-default-dim_rate_code",
  "schema": "",
  "table_type": "HIVE_TABLE",
  "status": "PUBLISHED",
  "tb_guid": null,
  "tb_logic_guid": "7beab16b-8464-4a9c-8b13-19eb997f8e2c",
  "dimension_name": "费率代码",
  "attributes": [ {
    "id": "1169323047937818624",
    "dimension_logic_table_id": "1169323047912652800",
    "ordinal": 1,
    "dimension_attribute_id": "1169323046453035008",
    "name_en": "rate_code_id",
    "name_ch": "费率id",
    "description": "",
    "data_type": "BIGINT",
    "domain_type": "NUMBER",
    "data_type_extend": null,
    "is_primary_key": true,
    "is_biz_primary": false,
    "is_partition_key": false,
    "not_null": false,
    "stand_row_id": null,
    "stand_row_name": null,
    "quality_infos": null,

```

```

    "alias": null,
    "self_defined_fields": [],
    "secrecy_levels": []
  }, {
    "id": "1169323047937818625",
    "dimension_logic_table_id": "1169323047912652800",
    "ordinal": 2,
    "dimension_attribute_id": "1169323046453035009",
    "name_en": "rate_code_value",
    "name_ch": "费率说明",
    "description": "",
    "data_type": "STRING",
    "domain_type": "STRING",
    "data_type_extend": null,
    "is_primary_key": false,
    "is_biz_primary": false,
    "is_partition_key": false,
    "not_null": false,
    "stand_row_id": null,
    "stand_row_name": null,
    "quality_infos": null,
    "alias": null,
    "self_defined_fields": [],
    "secrecy_levels": []
  }
],
"dw_type": "MRS_HIVE",
"dw_name": null,
"l1": "城市交通",
"l2": "公共维度/公共维度",
"l3": null,
"create_time": "2023-11-01T17:12:20+08:00",
"update_time": "2023-11-16T14:27:42+08:00",
"approval_info": null,
"new_biz": null,
"physical_table": "CREATE_SUCCESS",
"dev_physical_table": "NO_NEED",
"technical_asset": "CREATE_FAILED",
"business_asset": "UPDATE_SUCCESS",
"meta_data_link": "UPDATE_FAILED",
"data_quality": "NO_NEED",
"materialization": "NO_NEED",
"summary_status": "SUMMARY_FAILED",
"distribute": null,
"distribute_column": null,
"obs_location": null,
"quality_id": null,
"alias": "",
"configs": "{}",
"self_defined_fields": null,
"dev_version": null,
"prod_version": null,
"dev_version_name": null,
"prod_version_name": null,
"env_type": "INVALID_TYPE"
}
}
}
}

```

**状态码：400**

BadRequest

```

{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}

```

**状态码：401**

### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDimensionGroupsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDimensionGroupsRequest request = new ListDimensionGroupsRequest();
        try {
            ListDimensionGroupsResponse response = client.listDimensionGroups(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListDimensionGroupsRequest()
        response = client.list_dimension_groups(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListDimensionGroupsRequest{}
response, err := client.ListDimensionGroups(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是DerivativeIndexDimensionVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.15.7 查看逆向维度表任务

### 功能介绍

查看逆向维度表任务。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/dimension/database

表 8-995 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。



## 请求参数

表 8-996 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-997 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-998 data

参数	参数类型	描述
value	<a href="#">BatchOperationVO</a> object	value，统一的返回结果的外层数据结构。

表 8-999 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID, ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID, ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID, ID字符串。
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值: <ul style="list-style-type: none"> <li>• RUNNING: 运行中</li> <li>• SUCCESS: 操作成功</li> <li>• FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。
groups	Array of <a href="#">BatchOperationVO</a> objects	分组信息。

状态码: 400

表 8-1000 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码: 401

表 8-1001 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：403

表 8-1002 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

逆向数据库时，查看逆向任务。

```
GET https://{endpoint}/v2/{project_id}/design/dimension/database
```

## 响应示例

状态码：200

Success

```
{
  "data": {
    "value": {
      "id": null,
      "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70",
      "group_id": "1231306126000304128",
      "biz_name": null,
      "biz_id": null,
      "operation_status": "RUNNING",
      "operation_type": "DIMENSION_REVERSE_DB",
      "biz_info": null,
      "create_by": "test_uesr",
      "remark": null,
      "total": 1,
      "success": 0,
      "failed": 0,
      "rate": "0.00",
      "logs": null,
      "groups": null
    }
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CheckDimensionStatusSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CheckDimensionStatusRequest request = new CheckDimensionStatusRequest();
        try {
            CheckDimensionStatusResponse response = client.checkDimensionStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CheckDimensionStatusRequest()
    response = client.check_dimension_status(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CheckDimensionStatusRequest{}
    response, err := client.CheckDimensionStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.16 限定接口

### 8.16.1 查找业务限定

#### 功能介绍

通过中英文名称、描述、创建者、审核人、限定分组id、修改时间状态分页查找限定信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/conditions

表 8-1003 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1004 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。

参数	是否必选	参数类型	描述
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-1005 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1006 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1007 data

参数	参数类型	描述
value	<b>value</b> object	value，统一的返回结果的外层数据结构。

表 8-1008 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>ConditionVO</b> objects	ConditionVO信息。



表 8-1009 ConditionVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	字段名
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
condition_fn	String	限定计算方法。 枚举值： <ul style="list-style-type: none"> <li>• LAST_YEAR: 前一年</li> <li>• CURRENT_YEAR: 本年</li> <li>• BETWEEN_YEAR: 自定义年区间</li> <li>• LAST_MONTH: 前一月</li> <li>• CURRENT_MONTH: 本月</li> <li>• BETWEEN_MONTH: 自定义月区间</li> <li>• LAST_DAY: 前一天</li> <li>• CURRENT_DAY: 本日</li> <li>• BETWEEN_DAY: 自定义日区间</li> <li>• LAST_HOUR: 上一小时</li> <li>• CURRENT_HOUR: 当前小时</li> <li>• BETWEEN_HOUR: 自定义小时区间</li> <li>• LAST_MINUTE: 上一分钟</li> <li>• CURRENT_MINUTE: 当前分钟</li> <li>• BETWEEN_MINUTE: 自定义分钟区间</li> </ul>
condition_fn_param	String	限定计算参数。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

参数	参数类型	描述
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理, 只读。
base_time	Integer	基准时间。

表 8-1010 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1011 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-1012 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1013 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1014 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查找时间限定信息。

```
GET https://{endpoint}/v2/{project_id}/design/conditions?condition_type=TIME_CONDITION
```

## 响应示例

状态码：200

Success，返回data数据是ConditionVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 29,
      "records": [ {
        "id": "1204396387974955008",
        "name_en": "last_month",
```



```

"name_ch": "前一月",
"description": "",
"create_by": "test_uesr",
"condition_fn": "LAST_MONTH",
"condition_fn_param": "0",
"status": "PUBLISHED",
"create_time": "2024-02-06T12:01:15+08:00",
"update_time": "2024-03-07T18:01:43+08:00",
"approval_info": {
  "id": "1215358735950708736",
  "tenant_id": null,
  "name_ch": null,
  "name_en": null,
  "biz_id": "1204396387974955008",
  "biz_type": null,
  "biz_info": null,
  "biz_info_obj": null,
  "biz_version": 0,
  "biz_status": "PUBLISHED",
  "approval_status": "APPROVED",
  "approval_type": null,
  "submit_time": "2024-03-07T18:01:43+08:00",
  "create_by": null,
  "l1": null,
  "l2": null,
  "l3": null,
  "approval_time": "2024-03-07T18:01:43+08:00",
  "approver": "test_uesr",
  "email": null,
  "msg": "Fast Approval OK.",
  "directory_path": null
},
"new_biz": null,
"base_time": 1
}, {
  "id": "1167067126117302272",
  "name_en": "t003",
  "name_ch": "t003",
  "description": "",
  "create_by": "test_uesr",
  "condition_fn": "LAST_YEAR",
  "condition_fn_param": "0",
  "status": "PUBLISHED",
  "create_time": "2023-10-26T11:48:06+08:00",
  "update_time": "2023-11-09T13:36:28+08:00",
  "approval_info": {
    "id": "1172167829596860416",
    "tenant_id": null,
    "name_ch": null,
    "name_en": null,
    "biz_id": "1167067126117302272",
    "biz_type": null,
    "biz_info": null,
    "biz_info_obj": null,
    "biz_version": 0,
    "biz_status": "PUBLISHED",
    "approval_status": "APPROVED",
    "approval_type": null,
    "submit_time": "2023-11-09T13:36:28+08:00",
    "create_by": null,
    "l1": null,
    "l2": null,
    "l3": null,
    "approval_time": "2023-11-09T13:36:28+08:00",
    "approver": "test_uesr",
    "email": null,
    "msg": "Fast Approval OK. (Associated Publish By t003, Automatic Approval.)",
    "directory_path": null
  },

```

```

"new_biz" : null,
"base_time" : null
}, {
  "id" : "1163941657377488896",
  "name_en" : "t002",
  "name_ch" : "t002",
  "description" : "",
  "create_by" : "test_uesr",
  "condition_fn" : "LAST_YEAR",
  "condition_fn_param" : "0",
  "status" : "PUBLISHED",
  "create_time" : "2023-10-17T20:48:36+08:00",
  "update_time" : "2023-10-18T13:57:14+08:00",
  "approval_info" : {
    "id" : "1163941893843959808",
    "tenant_id" : null,
    "name_ch" : null,
    "name_en" : null,
    "biz_id" : "1163941657377488896",
    "biz_type" : null,
    "biz_info" : null,
    "biz_info_obj" : null,
    "biz_version" : 0,
    "biz_status" : "PUBLISHED",
    "approval_status" : "APPROVED",
    "approval_type" : null,
    "submit_time" : "2023-10-17T20:49:32+08:00",
    "create_by" : null,
    "l1" : null,
    "l2" : null,
    "l3" : null,
    "approval_time" : "2023-10-17T21:20:02+08:00",
    "approver" : "test_uesr",
    "email" : null,
    "msg" : "1",
    "directory_path" : null
  },
  "new_biz" : {
    "id" : null,
    "biz_type" : "TIME_CONDITION",
    "biz_id" : "1163941657377488896",
    "biz_info" : "{\"versionName\":null,\"versionTag\":null,\"versionId\":null,\"createBy\":null,\"updateBy\":null,\"createTime\":\"1697546916000\",\"updateTime\":\"1698292074309\",\"l3Id\":null,\"l2Id\":null,\"alias\":null,\"encoding\":null,\"prefix\":null,\"codeNum\":null,\"id\":\"1163941657377488896\",\"tenantId\":null,\"nameCh\":\"t002\",\"nameEn\":\"t002\",\"description\":\"\",\"conditionFn\":\"LAST_YEAR\",\"calExp\":null,\"conditionFnParam\":\"0\",\"status\":\"PUBLISHED\",\"enableFlag\":true,\"nameTranslate\":null,\"descriptionTranslate\":null,\"baseTime\":null,\"lastL2Id\":null}",
    "status" : "DRAFT",
    "biz_version" : 0,
    "create_time" : "2023-10-26T11:47:54+08:00",
    "update_time" : "2023-10-26T11:47:54+08:00"
  },
  "base_time" : null
}, {
  "id" : "1163941632085835776",
  "name_en" : "t001",
  "name_ch" : "t001",
  "description" : "",
  "create_by" : "test_uesr",
  "condition_fn" : "LAST_YEAR",
  "condition_fn_param" : "0",
  "status" : "OFFLINE",
  "create_time" : "2023-10-17T20:48:30+08:00",
  "update_time" : "2023-10-24T20:15:22+08:00",
  "approval_info" : null,
  "new_biz" : null,
  "base_time" : null
}

```

```
}, {
  "id": "1163884268984922112",
  "name_en": "sj1017",
  "name_ch": "sj1017",
  "description": "",
  "create_by": "test_uesr",
  "condition_fn": "LAST_YEAR",
  "condition_fn_param": "0",
  "status": "PUBLISHED",
  "create_time": "2023-10-17T17:00:34+08:00",
  "update_time": "2023-10-30T16:37:42+08:00",
  "approval_info": {
    "id": "1163926109700734976",
    "tenant_id": null,
    "name_ch": null,
    "name_en": null,
    "biz_id": "1163884268984922112",
    "biz_type": null,
    "biz_info": null,
    "biz_info_obj": null,
    "biz_version": 0,
    "biz_status": "PUBLISHED",
    "approval_status": "APPROVED",
    "approval_type": null,
    "submit_time": "2023-10-17T19:46:49+08:00",
    "create_by": null,
    "l1": null,
    "l2": null,
    "l3": null,
    "approval_time": "2023-10-17T19:50:00+08:00",
    "approver": "test_uesr",
    "email": null,
    "msg": "ok",
    "directory_path": null
  },
  "new_biz": null,
  "base_time": null
}, {
  "id": "1155167466507448320",
  "name_en": "lix",
  "name_ch": "lix",
  "description": "",
  "create_by": "test_uesr",
  "condition_fn": "LAST_MONTH",
  "condition_fn_param": "0",
  "status": "PUBLISHED",
  "create_time": "2023-09-23T15:43:06+08:00",
  "update_time": "2023-10-17T20:46:11+08:00",
  "approval_info": {
    "id": "1155919632490725376",
    "tenant_id": null,
    "name_ch": null,
    "name_en": null,
    "biz_id": "1155167466507448320",
    "biz_type": null,
    "biz_info": null,
    "biz_info_obj": null,
    "biz_version": 0,
    "biz_status": "PUBLISHED",
    "approval_status": "APPROVED",
    "approval_type": null,
    "submit_time": "2023-09-25T17:31:56+08:00",
    "create_by": null,
    "l1": null,
    "l2": null,
    "l3": null,
    "approval_time": "2023-09-25T17:31:56+08:00",
    "approver": "test_uesr",
    "email": null,
```

```

    "msg": "Fast Approval OK. (Associated Publish By lix, Automatic Approval.)",
    "directory_path": null
  },
  "new_biz": {
    "id": null,
    "biz_type": "TIME_CONDITION",
    "biz_id": "1155167466507448320",
    "biz_info": "{\versionName\":null,\versionTag\":null,\versionId\":null,\createBy
\\:\0833a573fe80d5401f6dc010a775569a\\,\updateBy\":\0833a573fe80d5401f6dc010a775569a
\\,\createTime\":1695454986000,\updateTime\":1698149390708,\l3Id\":null,\l2Id\":null,\alias
\\:\0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70\\,\nameCh
\\:\lix\\,\nameEn\":\lix\\,\description\":\\,\conditionFn\":\LAST_MONTH\\,\calExp
\\:\null\\,\conditionFnParam\":\0\\,\status\":\PUBLISHED\\,\enableFlag\":true,\nameTranslate
\\:\null\\,\descriptionTranslate\":null,\baseTime\":2,\lastL2Id\":null},
    "status": "DRAFT",
    "biz_version": 0,
    "create_time": "2023-10-24T20:09:51+08:00",
    "update_time": "2023-10-24T20:09:51+08:00"
  },
  "base_time": 2
}, {
  "id": "1154507394794024960",
  "name_en": "oi11",
  "name_ch": "oi11",
  "description": "",
  "create_by": "test_uesr",
  "condition_fn": "LAST_YEAR",
  "condition_fn_param": "0",
  "status": "PUBLISHED",
  "create_time": "2023-09-21T20:00:13+08:00",
  "update_time": "2023-10-17T16:54:20+08:00",
  "approval_info": {
    "id": "1163882705348386816",
    "tenant_id": null,
    "name_ch": null,
    "name_en": null,
    "biz_id": "1154507394794024960",
    "biz_type": null,
    "biz_info": null,
    "biz_info_obj": null,
    "biz_version": 0,
    "biz_status": "PUBLISHED",
    "approval_status": "APPROVED",
    "approval_type": null,
    "submit_time": "2023-10-17T16:54:21+08:00",
    "create_by": null,
    "l1": null,
    "l2": null,
    "l3": null,
    "approval_time": "2023-10-17T16:54:20+08:00",
    "approver": "test_uesr",
    "email": null,
    "msg": "Fast Approval OK.",
    "directory_path": null
  },
  "new_biz": null,
  "base_time": null
}, {
  "id": "1154507340083523584",
  "name_en": "lxy0101",
  "name_ch": "lxy0101",
  "description": "",
  "create_by": "test_uesr",
  "condition_fn": "LAST_YEAR",
  "condition_fn_param": "0",
  "status": "PUBLISHED",
  "create_time": "2023-09-21T20:00:00+08:00",
  "update_time": "2023-10-30T16:37:06+08:00",

```

```

"approval_info" : null,
"new_biz" : null,
"base_time" : null
}, {
  "id" : "1150823722467344384",
  "name_en" : "current_month",
  "name_ch" : "当前月",
  "description" : "",
  "create_by" : "test_uesr",
  "condition_fn" : "CURRENT_MONTH",
  "condition_fn_param" : "0",
  "status" : "PUBLISHED",
  "create_time" : "2023-09-11T16:02:37+08:00",
  "update_time" : "2023-09-11T16:02:37+08:00",
  "approval_info" : {
    "id" : "1150823723427840000",
    "tenant_id" : null,
    "name_ch" : null,
    "name_en" : null,
    "biz_id" : "1150823722467344384",
    "biz_type" : null,
    "biz_info" : null,
    "biz_info_obj" : null,
    "biz_version" : 0,
    "biz_status" : "PUBLISHED",
    "approval_status" : "APPROVED",
    "approval_type" : null,
    "submit_time" : "2023-09-11T16:02:37+08:00",
    "create_by" : null,
    "l1" : null,
    "l2" : null,
    "l3" : null,
    "approval_time" : "2023-09-11T16:02:37+08:00",
    "approver" : "test_uesr",
    "email" : null,
    "msg" : "Fast Approval OK.",
    "directory_path" : null
  },
  "new_biz" : null,
  "base_time" : 1
}, {
  "id" : "1128840997309136896",
  "name_en" : "aa",
  "name_ch" : "a",
  "description" : "",
  "create_by" : "test_uesr",
  "condition_fn" : "BETWEEN_DAY",
  "condition_fn_param" : "-1,1",
  "status" : "REJECT",
  "create_time" : "2023-07-13T00:11:07+08:00",
  "update_time" : "2024-02-27T21:44:53+08:00",
  "approval_info" : {
    "id" : "1200506282973483008",
    "tenant_id" : null,
    "name_ch" : null,
    "name_en" : null,
    "biz_id" : "1128840997309136896",
    "biz_type" : null,
    "biz_info" : null,
    "biz_info_obj" : null,
    "biz_version" : 0,
    "biz_status" : "REJECT",
    "approval_status" : "REJECT",
    "approval_type" : null,
    "submit_time" : "2024-01-26T18:23:22+08:00",
    "create_by" : null,
    "l1" : null,
    "l2" : null,
    "l3" : null,
  }
}

```

```

    "approval_time": "2024-02-27T21:44:53+08:00",
    "approver": "test_uesr",
    "email": null,
    "msg": "no",
    "directory_path": null
  },
  "new_biz": null,
  "base_time": 1
}, {
  "id": "1128832688892035072",
  "name_en": "last_year",
  "name_ch": "前一年",
  "description": "",
  "create_by": "test_uesr",
  "condition_fn": "LAST_YEAR",
  "condition_fn_param": "0",
  "status": "PUBLISHED",
  "create_time": "2023-07-12T23:38:06+08:00",
  "update_time": "2023-07-12T23:38:11+08:00",
  "approval_info": {
    "id": "1128832714003333120",
    "tenant_id": null,
    "name_ch": null,
    "name_en": null,
    "biz_id": "1128832688892035072",
    "biz_type": null,
    "biz_info": null,
    "biz_info_obj": null,
    "biz_version": 0,
    "biz_status": "PUBLISHED",
    "approval_status": "APPROVED",
    "approval_type": null,
    "submit_time": "2023-07-12T23:38:12+08:00",
    "create_by": null,
    "l1": null,
    "l2": null,
    "l3": null,
    "approval_time": "2023-07-12T23:38:11+08:00",
    "approver": "test_uesr",
    "email": null,
    "msg": "Fast Approval OK.",
    "directory_path": null
  },
  "new_biz": null,
  "base_time": null
}, {
  "id": "1128831725523312640",
  "name_en": "bbbb",
  "name_ch": "bbb",
  "description": "",
  "create_by": "test_uesr",
  "condition_fn": "LAST_YEAR",
  "condition_fn_param": "0",
  "status": "DRAFT",
  "create_time": "2023-07-12T23:34:16+08:00",
  "update_time": "2023-07-12T23:34:16+08:00",
  "approval_info": null,
  "new_biz": null,
  "base_time": null
}, {
  "id": "1128831552864788480",
  "name_en": "aaa",
  "name_ch": "aaa",
  "description": "",
  "create_by": "test_uesr",
  "condition_fn": "CURRENT_YEAR",
  "condition_fn_param": "0",
  "status": "DRAFT",
  "create_time": "2023-07-12T23:33:35+08:00",

```

```

"update_time" : "2023-07-12T23:34:33+08:00",
"approval_info" : null,
"new_biz" : null,
"base_time" : null
}, {
  "id" : "1109414797142806528",
  "name_en" : "1m_year_on_year",
  "name_ch" : "最近30天_去年同期",
  "description" : "",
  "create_by" : "test_uesr",
  "condition_fn" : "BETWEEN_DAY",
  "condition_fn_param" : "-395,-365",
  "status" : "PUBLISHED",
  "create_time" : "2023-05-20T09:38:20+08:00",
  "update_time" : "2023-05-20T09:38:19+08:00",
  "approval_info" : {
    "id" : "1109414797620957184",
    "tenant_id" : null,
    "name_ch" : null,
    "name_en" : null,
    "biz_id" : "1109414797142806528",
    "biz_type" : null,
    "biz_info" : null,
    "biz_info_obj" : null,
    "biz_version" : 0,
    "biz_status" : "PUBLISHED",
    "approval_status" : "APPROVED",
    "approval_type" : null,
    "submit_time" : "2023-05-20T09:38:20+08:00",
    "create_by" : null,
    "l1" : null,
    "l2" : null,
    "l3" : null,
    "approval_time" : "2023-05-20T09:38:19+08:00",
    "approver" : "test_uesr",
    "email" : null,
    "msg" : "Fast Approval OK.",
    "directory_path" : null
  },
  "new_biz" : {
    "id" : null,
    "biz_type" : "TIME_CONDITION",
    "biz_id" : "1109414797142806528",
    "biz_info" : "{\n  \"versionName\":null,\n  \"versionTag\":null,\n  \"versionId\":null,\n  \"createBy\":\n  \"0833a573fe80d5401f6dc010a775569a\", \n  \"updateBy\":null,\n  \"createTime\":1684546700000,\n  \"updateTime\":\n  1684546699000,\n  \"l3Id\":null,\n  \"l2Id\":null,\n  \"alias\":null,\n  \"encoding\":null,\n  \"prefix\":null,\n  \"codeNum\":null,\n  \"id\":\n  \"1109414797142806528\", \n  \"tenantId\":\n  \"0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70\", \n  \"nameCh\":\n  \"最近30天_去年同期\", \n  \"nameEn\":\n  \"1m_year_on_year\", \n  \"description\":\n  \"\", \n  \"conditionFn\":\n  \"BETWEEN_DAY\", \n  \"calExp\":null,\n  \"conditionFnParam\":\n  \"-395,-365\", \n  \"status\":\n  \"PUBLISHED\", \n  \"enableFlag\":true,\n  \"nameTranslate\":null,\n  \"descriptionTranslate\":null,\n  \"baseTime\":null,\n  \"lastL2Id\":null}",
    "status" : "DRAFT",
    "biz_version" : 0,
    "create_time" : "2023-06-01T14:17:10+08:00",
    "update_time" : "2023-06-01T14:17:10+08:00"
  },
  "base_time" : null
}, {
  "id" : "1109414655866064896",
  "name_en" : "1m_prior_period",
  "name_ch" : "最近30天_上期",
  "description" : "",
  "create_by" : "test_uesr",
  "condition_fn" : "BETWEEN_DAY",
  "condition_fn_param" : "-60,-30",
  "status" : "PUBLISHED",
  "create_time" : "2023-05-20T09:37:46+08:00",
  "update_time" : "2023-05-20T09:37:46+08:00",
  "approval_info" : {

```

```

    "id" : "1109414656310661120",
    "tenant_id" : null,
    "name_ch" : null,
    "name_en" : null,
    "biz_id" : "1109414655866064896",
    "biz_type" : null,
    "biz_info" : null,
    "biz_info_obj" : null,
    "biz_version" : 0,
    "biz_status" : "PUBLISHED",
    "approval_status" : "APPROVED",
    "approval_type" : null,
    "submit_time" : "2023-05-20T09:37:46+08:00",
    "create_by" : null,
    "l1" : null,
    "l2" : null,
    "l3" : null,
    "approval_time" : "2023-05-20T09:37:46+08:00",
    "approver" : "test_uesr",
    "email" : null,
    "msg" : "Fast Approval OK.",
    "directory_path" : null
  },
  "new_biz" : null,
  "base_time" : null
}, {
  "id" : "1013133487357812749",
  "name_en" : "f4w",
  "name_ch" : "未来4周",
  "description" : "历史数据的每天平均值*7*4",
  "create_by" : "SYSTEM",
  "condition_fn" : "BETWEEN_DAY",
  "condition_fn_param" : "1,28",
  "status" : "PUBLISHED",
  "create_time" : "2022-08-27T17:10:46+08:00",
  "update_time" : "2022-08-27T17:10:46+08:00",
  "approval_info" : null,
  "new_biz" : null,
  "base_time" : null
}, {
  "id" : "1013133487357812748",
  "name_en" : "f1w",
  "name_ch" : "未来7天",
  "description" : "历史数据的每天平均值*7",
  "create_by" : "SYSTEM",
  "condition_fn" : "BETWEEN_DAY",
  "condition_fn_param" : "1,7",
  "status" : "PUBLISHED",
  "create_time" : "2022-08-27T17:10:46+08:00",
  "update_time" : "2022-08-27T17:10:46+08:00",
  "approval_info" : null,
  "new_biz" : null,
  "base_time" : null
}, {
  "id" : "1013133487357812747",
  "name_en" : "1h",
  "name_ch" : "最近1小时",
  "description" : "上一个小时的数据",
  "create_by" : "SYSTEM",
  "condition_fn" : "BETWEEN_HOUR",
  "condition_fn_param" : "-1,-1",
  "status" : "PUBLISHED",
  "create_time" : "2022-08-27T17:10:46+08:00",
  "update_time" : "2022-08-27T17:10:46+08:00",
  "approval_info" : null,
  "new_biz" : null,
  "base_time" : null
}, {
  "id" : "1013133487357812746",

```



```

"name_en": "td",
"name_ch": "截止当日",
"description": "截止计算日期（包括）的所有数据(近10年内)",
"create_by": "SYSTEM",
"condition_fn": "BETWEEN_DAY",
"condition_fn_param": "-3650,0",
"status": "PUBLISHED",
"create_time": "2022-08-27T17:10:46+08:00",
"update_time": "2022-08-27T17:10:46+08:00",
"approval_info": null,
"new_biz": null,
"base_time": null
}, {
  "id": "1013133487357812745",
  "name_en": "b30d",
  "name_ch": "30天前的一天",
  "description": "往前第31天的数据",
  "create_by": "SYSTEM",
  "condition_fn": "BETWEEN_DAY",
  "condition_fn_param": "-31,-30",
  "status": "PUBLISHED",
  "create_time": "2022-08-27T17:10:46+08:00",
  "update_time": "2022-08-27T17:10:46+08:00",
  "approval_info": null,
  "new_biz": null,
  "base_time": null
}, {
  "id": "1013133487357812744",
  "name_en": "bh",
  "name_ch": "180天以前所有",
  "description": "截止180天前（往前第181天）的所有数据(近10年内), before half-year",
  "create_by": "SYSTEM",
  "condition_fn": "BETWEEN_DAY",
  "condition_fn_param": "-3650,-181",
  "status": "PUBLISHED",
  "create_time": "2022-08-27T17:10:46+08:00",
  "update_time": "2022-08-27T17:10:46+08:00",
  "approval_info": null,
  "new_biz": null,
  "base_time": null
}, {
  "id": "1013133487357812743",
  "name_en": "6m",
  "name_ch": "最近180天",
  "description": "截止昨天共180天的数据",
  "create_by": "SYSTEM",
  "condition_fn": "BETWEEN_DAY",
  "condition_fn_param": "-180,-1",
  "status": "PUBLISHED",
  "create_time": "2022-08-27T17:10:46+08:00",
  "update_time": "2022-08-27T17:10:46+08:00",
  "approval_info": null,
  "new_biz": null,
  "base_time": null
}, {
  "id": "1013133487357812742",
  "name_en": "3m",
  "name_ch": "最近90天",
  "description": "截止昨天共90天的数据",
  "create_by": "SYSTEM",
  "condition_fn": "BETWEEN_DAY",
  "condition_fn_param": "-90,-1",
  "status": "PUBLISHED",
  "create_time": "2022-08-27T17:10:46+08:00",
  "update_time": "2022-08-27T17:10:46+08:00",
  "approval_info": null,
  "new_biz": null,
  "base_time": null
}, {

```

```

"id" : "1013133487357812741",
"name_en" : "2m",
"name_ch" : "最近60天",
"description" : "截止昨天共60天的数据",
"create_by" : "SYSTEM",
"condition_fn" : "BETWEEN_DAY",
"condition_fn_param" : "-60,-1",
"status" : "PUBLISHED",
"create_time" : "2022-08-27T17:10:46+08:00",
"update_time" : "2022-08-27T17:10:46+08:00",
"approval_info" : null,
"new_biz" : null,
"base_time" : null
}, {
"id" : "1013133487357812740",
"name_en" : "1m",
"name_ch" : "最近30天",
"description" : "截止昨天共30天的数据",
"create_by" : "SYSTEM",
"condition_fn" : "BETWEEN_DAY",
"condition_fn_param" : "-30,-1",
"status" : "PUBLISHED",
"create_time" : "2022-08-27T17:10:46+08:00",
"update_time" : "2022-08-27T17:10:46+08:00",
"approval_info" : null,
"new_biz" : null,
"base_time" : null
}, {
"id" : "1013133487357812739",
"name_en" : "2w",
"name_ch" : "最近14天",
"description" : "截止昨天共14天的数据",
"create_by" : "SYSTEM",
"condition_fn" : "BETWEEN_DAY",
"condition_fn_param" : "-14,-1",
"status" : "PUBLISHED",
"create_time" : "2022-08-27T17:10:46+08:00",
"update_time" : "2022-08-27T17:10:46+08:00",
"approval_info" : null,
"new_biz" : null,
"base_time" : null
}, {
"id" : "1013133487357812738",
"name_en" : "1w",
"name_ch" : "最近7天",
"description" : "截止昨天共7天的数据",
"create_by" : "SYSTEM",
"condition_fn" : "BETWEEN_DAY",
"condition_fn_param" : "-7,-1",
"status" : "PUBLISHED",
"create_time" : "2022-08-27T17:10:46+08:00",
"update_time" : "2022-08-27T17:10:46+08:00",
"approval_info" : null,
"new_biz" : null,
"base_time" : null
}, {
"id" : "1013133487357812737",
"name_en" : "3d",
"name_ch" : "最近3天",
"description" : "截止昨天共3天的数据",
"create_by" : "SYSTEM",
"condition_fn" : "BETWEEN_DAY",
"condition_fn_param" : "-3,-1",
"status" : "PUBLISHED",
"create_time" : "2022-08-27T17:10:46+08:00",
"update_time" : "2022-08-27T17:10:46+08:00",
"approval_info" : null,
"new_biz" : null,
"base_time" : null

```

```
}, {
  "id": "1013133487357812736",
  "name_en": "1d",
  "name_ch": "最近1天",
  "description": "昨天的值",
  "create_by": "SYSTEM",
  "condition_fn": "BETWEEN_DAY",
  "condition_fn_param": "-1,-1",
  "status": "PUBLISHED",
  "create_time": "2022-08-27T17:10:46+08:00",
  "update_time": "2022-08-27T17:10:46+08:00",
  "approval_info": null,
  "new_biz": null,
  "base_time": null
}]
}
```

#### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

#### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

#### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListConditionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListConditionRequest request = new ListConditionRequest();
try {
    ListConditionResponse response = client.listCondition(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListConditionRequest()
        response = client.list_condition(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListConditionRequest{}
    response, err := client.ListCondition(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是ConditionVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.16.2 查看限定详情

### 功能介绍

通过ID查看限定详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/conditions/{id}

表 8-1015 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-1016 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

### 请求参数

表 8-1017 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-1018 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-1019 data

参数	参数类型	描述
value	<b>ConditionVO</b> object	value, 统一的返回结果的外层数据结构。

表 8-1020 ConditionVO

参数	参数类型	描述
id	String	编码, ID字符串。
name_en	String	字段名
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。

参数	参数类型	描述
condition_fn	String	<p>限定计算方法。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• LAST_YEAR: 前一年</li> <li>• CURRENT_YEAR: 本年</li> <li>• BETWEEN_YEAR: 自定义年区间</li> <li>• LAST_MONTH: 前一月</li> <li>• CURRENT_MONTH: 本月</li> <li>• BETWEEN_MONTH: 自定义月区间</li> <li>• LAST_DAY: 前一天</li> <li>• CURRENT_DAY: 本日</li> <li>• BETWEEN_DAY: 自定义日区间</li> <li>• LAST_HOUR: 上一小时</li> <li>• CURRENT_HOUR: 当前小时</li> <li>• BETWEEN_HOUR: 自定义小时区间</li> <li>• LAST_MINUTE: 上一分钟</li> <li>• CURRENT_MINUTE: 当前分钟</li> <li>• BETWEEN_MINUTE: 自定义分钟区间</li> </ul>
condition_fn_param	String	限定计算参数。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	<p>创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。</p>
update_time	String	<p>更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。</p>



参数	参数类型	描述
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理,只读。
base_time	Integer	基准时间。

表 8-1021 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1022 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-1023 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1024 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1025 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

通过ID查看限定详情信息。

```
GET https://{endpoint}/v2/{project_id}/design/conditions/873154196235517965
```

## 响应示例

状态码：200

Success，返回data数据是ConditionVO详情。

```
{
  "data" : {
    "value" : {
      "id" : "873154196235517965",
      "name_en" : "f4w",
      "name_ch" : "未来4周",
      "description" : "历史数据的每天平均值*7*4",

```

```

"create_by" : "SYSTEM",
"condition_fn" : "BETWEEN_DAY",
"condition_fn_param" : "1,28",
"status" : "PUBLISHED",
"create_time" : "2021-08-06T10:43:03+08:00",
"update_time" : "2021-08-06T10:43:03+08:00",
"approval_info" : null,
"new_biz" : null,
"base_time" : null
}
}
}

```

#### 状态码：400

BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

#### 状态码：401

Unauthorized

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}

```

#### 状态码：403

Forbidden

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowConditionByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
    }
}

```



```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowConditionByIdRequest request = new ShowConditionByIdRequest();
request.withId("{id}");
try {
    ShowConditionByIdResponse response = client.showConditionById(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowConditionByIdRequest()
        request.id = "{id}"
        response = client.show_condition_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowConditionByIdRequest{}
    request.Id = "{id}"
    response, err := client.ShowConditionById(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是ConditionVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.17 维度表接口

### 8.17.1 查找维度表

#### 功能介绍

通过中英文名称、创建者、审核人、状态、修改时间分页查找维度表信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/dimension-logic-tables

表 8-1026 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1027 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。
owner	否	String	按负责人查询。
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"><li>• DRAFT: 草稿</li><li>• PUBLISH_DEVELOPING: 发布待审批</li><li>• PUBLISHED: 已发布</li><li>• OFFLINE_DEVELOPING: 下线待审批</li><li>• OFFLINE: 已下线</li><li>• REJECT: 已驳回</li></ul>

参数	是否必选	参数类型	描述
sync_status	否	String	同步状态枚举。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 同步中</li> <li>• NO_NEED: 未同步</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> </ul>
sync_key	否	Array of strings	同步任务类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• BUSINESS_ASSET: 同步业务资产</li> <li>• DATA_QUALITY: 创建质量作业</li> <li>• TECHNICAL_ASSET: 同步技术资产</li> <li>• META_DATA_LINK: 资产关联</li> <li>• PHYSICAL_TABLE: 创建表（生产环境）</li> <li>• DEV_PHYSICAL_TABLE: 创建表（开发环境）</li> <li>• DLF_TASK: 创建数据开发作业</li> <li>• MATERIALIZATION: 数值落库（码表）</li> <li>• PUBLISH_TO_DLM: 发布数据服务API</li> <li>• SUMMARY_STATUS: 整体状态</li> </ul>
l2_id	否	String	主题域l2的ID，ID字符串。
dimension_id	否	String	依据维度ID查维度属性，ID字符串。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	是否必选	参数类型	描述
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。
biz_catalog_id	否	String	所属的业务分层的ID。
dimension_type	否	String	维度类型。 枚举值： <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>

## 请求参数

表 8-1028 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-1029 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-1030 data

参数	参数类型	描述
value	<b>value</b> object	value, 统一的返回结果的外层数据结构。

表 8-1031 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>DimensionLogicTableVO</b> objects	DimensionVO信息。

表 8-1032 DimensionLogicTableVO

参数	参数类型	描述
id	String	表ID
tb_name	String	表名称。
tb_logic_name	String	逻辑实体名。
l1_id	String	主题域分组ID, 只读, ID字符串。

参数	参数类型	描述
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象ID, ID字符串。
create_by	String	创建人。
description	String	描述。
dimension_id	String	所属维度ID, ID字符串。
owner	String	资产责任人。
dimension_type	String	维度类型。 枚举值: <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
code_table_id	String	引用码表ID (新建/更新“码表维度”时必须), ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表, 只读。
queue_name	String	dli数据连接执行sql所需的队列, 数据连接类型为DLI时必须。
dw_id	String	数据连接ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
db_name	String	库名。
tb_id	String	数据表ID, 只读。
schema	String	DWS类型需要。
table_type	String	表类型。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	String	表发布后，创建的数据目录技术资产 guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产 guid，只读，创建和更新时无需填写。
dimension_name	String	关联维度名称，只读。
attributes	Array of <a href="#">DimensionLogicTableAttributeVO</a> objects	字段属性。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



参数	参数类型	描述
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理,只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
quality_id	String	质量ID，ID字符串。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
obs_location	String	外表路径
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	Long	所属模型ID。
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1033 CodeTableVO

参数	参数类型	描述
id	String	码表ID，ID字符串。

参数	参数类型	描述
name_en	String	表名称，英文名。
name_ch	String	表名称，中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID，ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-1034 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID，ID字符串。

参数	参数类型	描述
code_table_id	String	所属码表ID（更新时必填），ID字符串。
ordinal	Integer	序号。
name_en	String	字段名，英文。
name_ch	String	字段名，中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-1035 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID，ID字符串。
fd_id	String	所属码表属性ID，ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-1036 DimensionLogicTableAttributeVO

参数	参数类型	描述
id	String	维度表ID, ID字符串。
dimension_logic_table_id	String	所属维表ID。
ordinal	Integer	序号
dimension_attribute_id	String	维度属性ID, ID字符串。
name_en	String	字段名, 只读。
name_ch	String	业务属性, 只读。
description	String	描述, 只读。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键, 只读。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否主键分区, 只读。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID, ID字符串。
stand_row_name	String	关联的数据标准名称, 只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息, 只读。
alias	String	别名。
self_defined_fields	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级

表 8-1037 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1038 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理, 只读。

表 8-1039 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数 “project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1040 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-1041 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-1042 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1043 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1044 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1045 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

根据查询条件分页查找维度表。

GET https://{endpoint}/v2/{project\_id}/design/dimension-logic-tables?offset=1&limit=10

## 响应示例

**状态码：200**

Success, 返回data数据是DimensionLogicTableVO数组, 以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1222930979413499904",
        "tb_name": "dim_test_czh_0328",
        "tb_logic_name": "test_czh_0328",
        "l1_id": "1194966710038474752",
        "l2_id": null,
        "l3_id": null,
        "create_by": "test_uesr",
        "description": "无",
        "dimension_id": "1222930977286987776",
        "owner": "afcaca",
        "dimension_type": "COMMON",
        "code_table_id": null,
        "code_table": null,
        "queue_name": "default",
        "dw_id": "8259af52bd294f98920ebce75a426391",
        "reversed": null,
        "partition_conf": null,
        "dirty_out_switch": false,
        "dirty_out_database": "",
        "dirty_out_prefix": "",
        "dirty_out_suffix": "",
        "db_name": "bi",
        "tb_id": null,
        "schema": "",
        "table_type": "MANAGED",
        "status": "PUBLISHED",
        "tb_guid": null,
        "tb_logic_guid": "712edcdd-3bba-4077-b182-4a1f17e5bcd8",
        "dimension_name": "test_czh_0328",
        "attributes": [ {
          "id": "1222930979442860032",
          "dimension_logic_table_id": "1222930979413499904",
          "ordinal": 1,
          "dimension_attribute_id": "1222930977370873856",
          "name_en": "dim_test_czh_0328_key",
          "name_ch": "test_czh_0328",
          "description": "",
          "data_type": "BIGINT",
          "domain_type": "NUMBER",
          "data_type_extend": null,
          "is_primary_key": true,
          "is_biz_primary": false,
          "is_partition_key": false,

```

```

"not_null" : false,
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : ""
} ],
"secrecy_levels" : [ ]
}, {
  "id" : "1222930979442860033",
  "dimension_logic_table_id" : "1222930979413499904",
  "ordinal" : 2,
  "dimension_attribute_id" : "1222930977370873857",
  "name_en" : "aaa",
  "name_ch" : "aaa",
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : null,
  "is_primary_key" : false,
  "is_biz_primary" : false,
  "is_partition_key" : false,
  "not_null" : false,
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "aa",
    "fd_name_en" : "aa",
    "not_null" : false,
    "fd_value" : ""
  }, {
    "fd_name_ch" : "bb",
    "fd_name_en" : "bb",
    "not_null" : false,
    "fd_value" : ""
  } ],
  "secrecy_levels" : [ ]
}, {
  "id" : "1222930979442860034",
  "dimension_logic_table_id" : "1222930979413499904",
  "ordinal" : 3,
  "dimension_attribute_id" : "1222930977370873858",
  "name_en" : "bbb",
  "name_ch" : "bbb",
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : null,
  "is_primary_key" : false,
  "is_biz_primary" : false,
  "is_partition_key" : false,
  "not_null" : false,
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "aa",

```

```

        "fd_name_en" : "aa",
        "not_null" : false,
        "fd_value" : ""
    }, {
        "fd_name_ch" : "bb",
        "fd_name_en" : "bb",
        "not_null" : false,
        "fd_value" : ""
    }
    ],
    "secrecy_levels" : [ ]
}],
"dw_type" : "DWS",
"dw_name" : null,
"l1" : "test_czh_catalog",
"l2" : null,
"l3" : null,
"create_time" : "2024-03-28T15:31:06+08:00",
"update_time" : "2024-03-28T15:31:10+08:00",
"approval_info" : {
    "id" : "1222930978226511872",
    "tenant_id" : null,
    "name_ch" : null,
    "name_en" : null,
    "biz_id" : "1222930977286987776",
    "biz_type" : null,
    "biz_info" : null,
    "biz_info_obj" : null,
    "biz_version" : 0,
    "biz_status" : "PUBLISHED",
    "approval_status" : "APPROVED",
    "approval_type" : null,
    "submit_time" : "2024-03-28T15:31:06+08:00",
    "create_by" : null,
    "l1" : null,
    "l2" : null,
    "l3" : null,
    "approval_time" : "2024-03-28T15:31:06+08:00",
    "approver" : "test_uesr",
    "email" : null,
    "msg" : "Fast Approval OK.",
    "directory_path" : null
},
"new_biz" : null,
"physical_table" : "CREATE_FAILED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "NO_NEED",
"materialization" : null,
"summary_status" : "SUMMARY_FAILED",
"distribute" : null,
"distribute_column" : null,
"obs_location" : null,
"quality_id" : null,
"alias" : "",
"configs" : "{}",
"self_defined_fields" : [ {
    "fd_name_ch" : "自定义项1",
    "fd_name_en" : "selfDefine1",
    "not_null" : false,
    "fd_value" : ""
}, {
    "fd_name_ch" : "自定义项2",
    "fd_name_en" : "selfDefine2",
    "not_null" : true,
    "fd_value" : "adada"
}
],
"dev_version" : null,

```

```
"prod_version" : null,  
"dev_version_name" : null,  
"prod_version_name" : null,  
"env_type" : "PROD_TYPE"  
  }]  
}  
}
```

**状态码：400**

BadRequest

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user request is illegal."  
}
```

**状态码：401**

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

**状态码：403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListDimensionLogicTablesSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)
```

```

        .withAk(ak)
        .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListDimensionLogicTablesRequest request = new ListDimensionLogicTablesRequest();
try {
    ListDimensionLogicTablesResponse response = client.listDimensionLogicTables(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

## Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListDimensionLogicTablesRequest()
        response = client.list_dimension_logic_tables(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"

```

```

dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDimensionLogicTablesRequest{}
    response, err := client.ListDimensionLogicTables(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是DimensionLogicTableVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.17.2 删除维度表

### 功能介绍

根据维度表ID，删除维度表。



## 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v2/{project\_id}/design/dimension-logic-tables

表 8-1046 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-1047 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1048 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。

参数	是否必选	参数类型	描述
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-1049 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-1050 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-1051 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1052 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-1053 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据传入的参数，删除维度表，只支持删除状态为草稿、已下线、已驳回的维度表。

```
DELETE https://{endpoint}/v2/{project_id}/design/atomic-indexs
{
  "ids" : [ "1227990857618227200" ]
}
```

## 响应示例

**状态码：200**

Success，返回data数据是删除个数。

```
{
  "value" : 1
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

### Forbidden

```
{  
  "error_code": "DS.60xx",  
  "error_msg": "The user does not have permission to call this API."  
}
```

## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.17.3 查看维度表详情

### 功能介绍

通过ID查看维度表的详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/dimension-logic-tables/{id}

表 8-1054 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID, ID字符串。

表 8-1055 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

## 请求参数

表 8-1056 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1057 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-1058 data

参数	参数类型	描述
value	<a href="#">DimensionLogicT</a> <a href="#">ableVO</a> object	value，统一的返回结果的外层数据结构。

表 8-1059 DimensionLogicTableVO

参数	参数类型	描述
id	String	表ID
tb_name	String	表名称。
tb_logic_name	String	逻辑实体名。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
create_by	String	创建人。
description	String	描述。
dimension_id	String	所属维度ID，ID字符串。
owner	String	资产责任人。
dimension_type	String	维度类型。 枚举值： <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
code_table_id	String	引用码表ID（新建/更新“码表维度”时必须填），ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表，只读。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
dw_id	String	数据连接ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
db_name	String	库名。
tb_id	String	数据表ID，只读。

参数	参数类型	描述
schema	String	DWS类型需要。
table_type	String	表类型。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	String	表发布后，创建的数据目录技术资产guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产guid，只读，创建和更新时无需填写。
dimension_name	String	关联维度名称，只读。
attributes	Array of <a href="#">DimensionLogicTableAttributeVO</a> objects	字段属性。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
quality_id	String	质量ID，ID字符串。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
obs_location	String	外表路径
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	Long	所属模型ID。
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1060 CodeTableVO

参数	参数类型	描述
id	String	码表ID，ID字符串。

参数	参数类型	描述
name_en	String	表名称, 英文名。
name_ch	String	表名称, 中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-1061 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。

参数	参数类型	描述
code_table_id	String	所属码表ID（更新时必填），ID字符串。
ordinal	Integer	序号。
name_en	String	字段名，英文。
name_ch	String	字段名，中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <a href="#">CodeTableFieldValueVO</a> objects	码表属性值。
count_field_values	Integer	码表属性值总数。

表 8-1062 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID，ID字符串。
fd_id	String	所属码表属性ID，ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-1063 DimensionLogicTableAttributeVO

参数	参数类型	描述
id	String	维度表ID, ID字符串。
dimension_logic_table_id	String	所属维表ID。
ordinal	Integer	序号
dimension_attribute_id	String	维度属性ID, ID字符串。
name_en	String	字段名, 只读。
name_ch	String	业务属性, 只读。
description	String	描述, 只读。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键, 只读。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否主键分区, 只读。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID, ID字符串。
stand_row_name	String	关联的数据标准名称, 只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息, 只读。
alias	String	别名。
self_defined_fields	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级

表 8-1064 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1065 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理, 只读。

表 8-1066 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数 “project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1067 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1068 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-1069 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。



参数	参数类型	描述
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1070 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1071 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1072 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

通过ID查看维度表的详情信息。

GET [https://{endpoint}/v2/{project\\_id}/design/dimension-logic-tables/1222930979413499904](https://{endpoint}/v2/{project_id}/design/dimension-logic-tables/1222930979413499904)

## 响应示例

**状态码：200**

Success, 返回data数据是DimensionLogicTableVO详情。

```
{
  "data": {
    "value": {
      "id": "1222930979413499904",
      "tb_name": "dim_test_czh_0328",
      "tb_logic_name": "test_czh_0328",
      "l1_id": "1194966710038474752",
      "l2_id": null,
      "l3_id": null,
      "create_by": "test_uesr",
      "description": "无",
      "dimension_id": "1222930977286987776",
      "owner": "afcaca",
      "dimension_type": "COMMON",
      "code_table_id": null,
      "code_table": null,
      "queue_name": "default",
      "dw_id": "8259af52bd294f98920ebce75a426391",
      "reversed": null,
      "partition_conf": null,
      "dirty_out_switch": false,
      "dirty_out_database": "",
      "dirty_out_prefix": "",
      "dirty_out_suffix": "",
      "db_name": "bi",
      "tb_id": null,
      "schema": "",
      "table_type": "MANAGED",
      "status": "PUBLISHED",
      "tb_guid": null,
      "tb_logic_guid": "712edcdd-3bba-4077-b182-4a1f17e5bcd8",
      "dimension_name": "test_czh_0328",
      "attributes": [ {
        "id": "1222930979442860032",
        "dimension_logic_table_id": "1222930979413499904",
        "ordinal": 1,
        "dimension_attribute_id": "1222930977370873856",
        "name_en": "dim_test_czh_0328_key",
        "name_ch": "test_czh_0328",
        "description": "",
        "data_type": "BIGINT",
        "domain_type": "NUMBER",
        "data_type_extend": null,
        "is_primary_key": true,
        "is_biz_primary": false,
        "is_partition_key": false,
        "not_null": false,
        "stand_row_id": null,
      } ]
    }
  }
}
```

```

"stand_row_name" : null,
"quality_infos" : [ ],
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : ""
} ],
"secrecy_levels" : [ ]
}, {
  "id" : "1222930979442860033",
  "dimension_logic_table_id" : "1222930979413499904",
  "ordinal" : 2,
  "dimension_attribute_id" : "1222930977370873857",
  "name_en" : "aaa",
  "name_ch" : "aaa",
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : null,
  "is_primary_key" : false,
  "is_biz_primary" : false,
  "is_partition_key" : false,
  "not_null" : false,
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : [ ],
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "aa",
    "fd_name_en" : "aa",
    "not_null" : false,
    "fd_value" : ""
  }, {
    "fd_name_ch" : "bb",
    "fd_name_en" : "bb",
    "not_null" : false,
    "fd_value" : ""
  } ],
  "secrecy_levels" : [ ]
}, {
  "id" : "1222930979442860034",
  "dimension_logic_table_id" : "1222930979413499904",
  "ordinal" : 3,
  "dimension_attribute_id" : "1222930977370873858",
  "name_en" : "bbb",
  "name_ch" : "bbb",
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : null,
  "is_primary_key" : false,
  "is_biz_primary" : false,
  "is_partition_key" : false,
  "not_null" : false,
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : [ ],
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "aa",
    "fd_name_en" : "aa",
    "not_null" : false,

```

```

        "fd_value" : ""
    }, {
        "fd_name_ch" : "bb",
        "fd_name_en" : "bb",
        "not_null" : false,
        "fd_value" : ""
    }
    ],
    "secrecy_levels" : [ ]
}],
"dw_type" : "DWS",
"dw_name" : "dws",
"l1" : "test_czh_catalog",
"l2" : null,
"l3" : null,
"create_time" : "2024-03-28T15:31:06+08:00",
"update_time" : "2024-03-28T15:31:10+08:00",
"approval_info" : null,
"new_biz" : null,
"physical_table" : "CREATE_FAILED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "NO_NEED",
"materialization" : null,
"summary_status" : "SUMMARY_FAILED",
"distribute" : null,
"distribute_column" : null,
"obs_location" : null,
"quality_id" : null,
"alias" : "",
"configs" : "{}",
"self_defined_fields" : [ {
    "fd_name_ch" : "自定义项1",
    "fd_name_en" : "selfDefine1",
    "not_null" : false,
    "fd_value" : ""
}, {
    "fd_name_ch" : "自定义项2",
    "fd_name_en" : "selfDefine2",
    "not_null" : true,
    "fd_value" : "adada"
}
],
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE"
}
}
}

```

### 状态码：400

#### BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

### 状态码：401

#### Unauthorized

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}

```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

**SDK 代码示例**

SDK代码示例如下。

**Java**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowDimensionLogicTableByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDimensionLogicTableByIdRequest request = new ShowDimensionLogicTableByIdRequest();
        request.withId("{id}");
        try {
            ShowDimensionLogicTableByIdResponse response = client.showDimensionLogicTableById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowDimensionLogicTableByIdRequest()
        request.id = "{id}"
        response = client.show_dimension_logic_table_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```

        WithCredential(auth).
        Build()

        request := &model.ShowDimensionLogicTableByIdRequest{}
        request.Id = "{id}"
        response, err := client.ShowDimensionLogicTableById(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是DimensionLogicTableVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.18 事实表接口

### 8.18.1 查找事实表

#### 功能介绍

通过中英文名称、创建者、审核人、状态、修改时间分页查找事实表信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/fact-logic-tables

表 8-1073 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1074 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
name_ch	否	String	按名称精确查询。
name_en	否	String	按编码精确查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。
owner	否	String	按负责人查询。
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
sync_status	否	String	同步状态枚举。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 同步中</li> <li>• NO_NEED: 未同步</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> </ul>



参数	是否必选	参数类型	描述
sync_key	否	Array of strings	同步任务类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• BUSINESS_ASSET: 同步业务资产</li> <li>• DATA_QUALITY: 创建质量作业</li> <li>• TECHNICAL_ASSET: 同步技术资产</li> <li>• META_DATA_LINK: 资产关联</li> <li>• PHYSICAL_TABLE: 创建表（生产环境）</li> <li>• DEV_PHYSICAL_TABLE: 创建表（开发环境）</li> <li>• DLF_TASK: 创建数据开发作业</li> <li>• MATERIALIZATION: 数值落库（码表）</li> <li>• PUBLISH_TO_DLM: 发布数据服务API</li> <li>• SUMMARY_STATUS: 整体状态</li> </ul>
l3_id	否	String	业务对象l3的ID，ID字符串。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

参数	是否必选	参数类型	描述
biz_catalog_id	否	String	所属的业务分层的ID。

## 请求参数

表 8-1075 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1076 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1077 data

参数	参数类型	描述
value	<b>value</b> object	value，统一的返回结果的外层数据结构。

表 8-1078 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <a href="#">FactLogicTableVO</a> objects	FactLogicTableVO信息。

表 8-1079 FactLogicTableVO

参数	参数类型	描述
id	String	事实表ID, ID字符串。
tb_name	String	表名称。
tb_logic_name	String	逻辑实体名。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象guid, ID字符串。
description	String	描述。
dw_name	String	数据连接名称, 只读, 创建和更新时无需填写。
owner	String	资产责任人。
create_by	String	创建人。
queue_name	String	dli数据连接执行sql所需的队列, 数据连接类型为DLI时必须。
dw_id	String	数据连接ID。
db_name	String	库名。
tb_id	String	数据表ID, 只读。
dim_table_ids	Array of strings	关联维度表ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。

参数	参数类型	描述
dirty_out_suffix	String	异常表后缀。
schema	String	DWS类型需要。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
table_type	String	表类型。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	String	表发布后，创建的数据目录技术资产guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产guid，只读，创建和更新时无需填写。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理, 只读。
dimensions	Array of <b>FactTableAttributeVO</b> objects	事实表维度信息, 只读。
attributes	Array of <b>FactTableAttributeVO</b> objects	事实表事实属性信息, 只读。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
measures	Array of <b>FactTableAttributeVO</b> objects	事实表度量信息, 只读。
table_attributes	Array of <b>FactTableAttributeVO</b> objects	事实表所有属性。
configs	String	其他配置
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
quality_id	String	质量ID, ID字符串。
alias	String	别名。

参数	参数类型	描述
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
obs_location	String	外表路径
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
model_id	Long	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1080 FactTableAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
fact_logic_table_id	String	所属事实表ID，只读，ID字符串。
ordinal	Integer	序号。
dimension_id	String	维度ID，ID字符串。
dimension_attr_id	String	维度属性ID，ID字符串。
role	String	维度角色。
dimension	<b>DimensionVO</b> object	dimension_id对应的维度信息，只读。
is_primary_key	Boolean	是否主键。



参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键，只读。
secrecy_levels	Array of <a href="#">SecrecyLevelRecordVO</a> objects	密级
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
name_en	String	英文名。
name_ch	String	中文名。
not_null	Boolean	是否不为空。

参数	参数类型	描述
attribute_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。

表 8-1081 DimensionVO

参数	参数类型	描述
id	String	编码（新建时只读，更新时必填），ID字符串。
name_en	String	维度英文名称。
dimension_type	String	维度类型（更新时只能由普通维度改为层级维度，其余场景都不允许修改）。 枚举值： <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
name_ch	String	维度名称。

参数	参数类型	描述
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
code_table_id	String	引用码表ID（新建/更新“码表维度”时必须填），ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表，只读。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
hierarchies	Array of <b>DimensionHierar chiesVO</b> objects	层级属性。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	<b>BizDatasourceRelationVO</b> object	数据源信息，新建/更新时必填。
owner	String	资产责任人，新建/更新时必填。
obs_location	String	外表路径
table_type	String	表类型。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。

参数	参数类型	描述
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	String	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1082 CodeTableVO

参数	参数类型	描述
id	String	码表ID, ID字符串。
name_en	String	表名称, 英文名。
name_ch	String	表名称, 中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>DRAFT: 草稿</li> <li>PUBLISH_DEVELOPING: 发布待审核</li> <li>PUBLISHED: 已发布</li> <li>OFFLINE_DEVELOPING: 下线待审核</li> <li>OFFLINE: 已下线</li> <li>REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-1083 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <b>CodeTableFieldValueVO</b> objects	码表属性值。

参数	参数类型	描述
count_field_values	Integer	码表属性值总数。

表 8-1084 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-1085 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码, ID字符串。
name	String	层级名称。
attrs	Array of <a href="#">HierarchiesAttrVO</a> objects	层级包含的属性。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1086 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码, ID字符串。
hierarchies_id	String	层级ID, ID字符串。



参数	参数类型	描述
attr_id	String	属性ID, ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称, 只读。
detail_attr_ids	Array of strings	详情属性ID, ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文, 只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段, 只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段, 只读。

表 8-1087 DimensionAttributeVO

参数	参数类型	描述
id	String	编码, ID字符串。
dimension_id	String	维度ID, 只读, ID字符串。
code_table_field_id	String	码表属性ID, ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>

参数	参数类型	描述
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID, ID字符串。
stand_row_name	String	关联的数据标准名称, 只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息, 只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。

表 8-1088 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1089 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。

参数	参数类型	描述
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1090 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。
target_attr_id	String	目的字段ID，ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组，只读。
src_table_db_names	Array of strings	源表db名称数组，只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组，只读，ID字符串。
src_table_id_list	Array of strings	源表ID数组，只读，ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组，只读。
src_attr_id_list	Array of strings	源表字段ID数组，只读，ID字符串。
remark	String	备注。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-1091 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID，ID字符串。
table2_id	String	表2ID，ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1092 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID，ID字符串。
field2_id	String	属性2ID，ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。



表 8-1093 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-1094 BizDatasourceRelationVO

参数	参数类型	描述
id	String	编码, ID字符串。
biz_id	String	业务对象信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	String	数据连接ID。
dw_name	String	数据连接名，只读。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。

表 8-1095 SecrecyLevelRecordVO

参数	参数类型	描述
id	String	属性关联密级的ID，ID字符串。
secrecyLevel_id	String	密级的ID，ID字符串。
secrecyLevel_name	String	密级名称。
uuid	String	数据安全主键。
slevel	Integer	密级等级。
description	String	密级描述。

参数	参数类型	描述
biz_id	String	业务对象ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建者。
update_by	String	更新者。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1096 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-1097 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-1098 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<a href="#">BizVersionManageVO</a> object	业务版本管理, 只读。

表 8-1099 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1100 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

**状态码：400**

表 8-1101 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1102 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1103 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

通过查询条件获取事实表列表。

```
GET https://{endpoint}/v2/{project_id}/design/fact-logic-tables?
offset=1&limit=10&name=fact_test_czh_0305_003
```

## 响应示例

**状态码：200**

Success，返回data数据是FactLogicTableVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1214584210761940992",
        "tb_name": "fact_test_czh_0305_003",
        "tb_logic_name": "test_czh_0305_003",
        "l1_id": "1194966710038474752",
        "l2_id": null,
        "l3_id": null,
        "description": "无",
        "dw_name": null,
        "owner": "aaa",
        "create_by": "test_uesr",
        "queue_name": "default",
        "dw_id": "8259af52bd294f98920ebce75a426391",
        "db_name": "bi",
        "tb_id": "NativeTable-8259af52bd294f98920ebce75a426391-bi-fact_test_czh_0305_003",
        "dim_table_ids": null,
        "reversed": null,
        "partition_conf": null,
        "dirty_out_switch": false,
        "dirty_out_database": "",
        "dirty_out_prefix": "",
        "dirty_out_suffix": "",
        "schema": "",
        "distribute": null,
        "distribute_column": null,
        "obs_location": null,
        "table_type": "MANAGED",
        "status": "PUBLISHED",
        "tb_guid": null,
        "tb_logic_guid": "5417940a-0d1d-4fea-9b6f-65637da15112",
        "dw_type": "DWS",
        "l1": "test_czh_catalog",
        "l2": null,
        "l3": null,
        "create_time": "2024-03-05T14:44:02+08:00",
        "update_time": "2024-03-05T15:35:48+08:00",
        "approval_info": {
          "id": "1214597210327646208",
          "tenant_id": null,
          "name_ch": null,
          "name_en": null,
          "biz_id": "1214584210761940992",
          "biz_type": null,
          "biz_info": null,
          "biz_info_obj": null,
          "biz_version": 0,
          "biz_status": "PUBLISHED",
          "approval_status": "APPROVED",
          "approval_type": null,
          "submit_time": "2024-03-05T15:35:41+08:00",
          "create_by": null,
          "l1": null,
          "l2": null,
          "l3": null,
          "approval_time": "2024-03-05T15:35:41+08:00",
          "approver": "test_uesr",
          "email": null,
          "msg": "Fast Approval OK.",
          "directory_path": null
        },
        "new_biz": null,
        "dimensions": null,
        "attributes": null,
        "mappings": null,
        "measures": [ {
```

```

"id" : "1214584210866798592",
"fact_logic_table_id" : "1214584210761940992",
"ordinal" : 1,
"dimension_id" : null,
"role" : null,
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : null,
"secrecy_levels" : [ ],
"description" : "",
"data_type" : "BIGINT",
"domain_type" : "NUMBER",
"data_type_extend" : "",
"name_en" : "aaa",
"name_ch" : "aaa",
"not_null" : false,
"attribute_type" : "FACT_MEASURE",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : null
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : null
}
], {
  "id" : "1214584210866798593",
  "fact_logic_table_id" : "1214584210761940992",
  "ordinal" : 2,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ {
    "id" : "187",
    "secrecyLevel_id" : "1199020797063208960",
    "secrecyLevel_name" : "保密",
    "uuid" : "8a9481ba8c202835018c237ae6540007",
    "slevel" : 5,
    "description" : "答应我，一定不要对别人说哦。",
    "biz_id" : "1214584210866798593",
    "biz_type" : "FACT_LOGIC_TABLE",
    "create_by" : "test_uesr",
    "update_by" : "test_uesr",
    "create_time" : "2024-03-05T15:35:41+08:00",
    "update_time" : "2024-03-05T15:35:41+08:00"
  }
],
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : "",
  "name_en" : "bbb",
  "name_ch" : "bbb",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",

```

```

"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : null
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : null
} ]
}, {
  "id" : "1214584210866798594",
  "fact_logic_table_id" : "1214584210761940992",
  "ordinal" : 3,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ {
    "id" : "186",
    "secrecyLevel_id" : "1199020797063208960",
    "secrecyLevel_name" : "保密",
    "uuid" : "8a9481ba8c202835018c237ae6540007",
    "slevel" : 5,
    "description" : "答应我，一定不要对别人说哦。",
    "biz_id" : "1214584210866798594",
    "biz_type" : "FACT_LOGIC_TABLE",
    "create_by" : "test_uesr",
    "update_by" : "test_uesr",
    "create_time" : "2024-03-05T15:35:41+08:00",
    "update_time" : "2024-03-05T15:35:41+08:00"
  } ],
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : "",
  "name_en" : "ccc",
  "name_ch" : "ccc",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "aa",
    "fd_name_en" : "aa",
    "not_null" : false,
    "fd_value" : null
  }, {
    "fd_name_ch" : "bb",
    "fd_name_en" : "bb",
    "not_null" : false,
    "fd_value" : null
  } ]
} ],
"table_attributes" : [ {
  "id" : "1214584210866798592",
  "fact_logic_table_id" : "1214584210761940992",
  "ordinal" : 1,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,

```



```

"secrecy_levels" : [ {
  "id" : "188",
  "secrecyLevel_id" : "1199020797063208960",
  "secrecyLevel_name" : "保密",
  "uuid" : "8a9481ba8c202835018c237ae6540007",
  "slevel" : 5,
  "description" : "答应我，一定不要对别人说哦。",
  "biz_id" : "1214584210866798592",
  "biz_type" : "FACT_LOGIC_TABLE",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-03-05T15:35:41+08:00",
  "update_time" : "2024-03-05T15:35:41+08:00"
} ],
"description" : "",
"data_type" : "BIGINT",
"domain_type" : "NUMBER",
"data_type_extend" : "",
"name_en" : "aaa",
"name_ch" : "aaa",
"not_null" : false,
"attribute_type" : "FACT_MEASURE",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : null
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : null
} ]
}, {
  "id" : "1214584210866798593",
  "fact_logic_table_id" : "1214584210761940992",
  "ordinal" : 2,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ {
    "id" : "187",
    "secrecyLevel_id" : "1199020797063208960",
    "secrecyLevel_name" : "保密",
    "uuid" : "8a9481ba8c202835018c237ae6540007",
    "slevel" : 5,
    "description" : "答应我，一定不要对别人说哦。",
    "biz_id" : "1214584210866798593",
    "biz_type" : "FACT_LOGIC_TABLE",
    "create_by" : "test_uesr",
    "update_by" : "test_uesr",
    "create_time" : "2024-03-05T15:35:41+08:00",
    "update_time" : "2024-03-05T15:35:41+08:00"
  } ],
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : "",
  "name_en" : "bbb",
  "name_ch" : "bbb",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",

```

```

"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : null
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : null
} ]
}, {
  "id" : "1214584210866798594",
  "fact_logic_table_id" : "1214584210761940992",
  "ordinal" : 3,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ {
    "id" : "186",
    "secrecyLevel_id" : "1199020797063208960",
    "secrecyLevel_name" : "保密",
    "uuid" : "8a9481ba8c202835018c237ae6540007",
    "slevel" : 5,
    "description" : "答应我，一定不要对别人说哦。",
    "biz_id" : "1214584210866798594",
    "biz_type" : "FACT_LOGIC_TABLE",
    "create_by" : "test_uesr",
    "update_by" : "test_uesr",
    "create_time" : "2024-03-05T15:35:41+08:00",
    "update_time" : "2024-03-05T15:35:41+08:00"
  } ],
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",
  "data_type_extend" : "",
  "name_en" : "ccc",
  "name_ch" : "ccc",
  "not_null" : false,
  "attribute_type" : "FACT_MEASURE",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : null,
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "aa",
    "fd_name_en" : "aa",
    "not_null" : false,
    "fd_value" : null
  }, {
    "fd_name_ch" : "bb",
    "fd_name_en" : "bb",
    "not_null" : false,
    "fd_value" : null
  } ]
} ],
"physical_table" : "UPDATE_SUCCESS",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_FAILED",
"business_asset" : "UPDATE_SUCCESS",
"meta_data_link" : "UPDATE_FAILED",
"data_quality" : "NO_NEED",

```

```

"summary_status" : "SUMMARY_FAILED",
"quality_id" : null,
"alias" : "",
"configs" : "{}",
"self_defined_fields" : [ {
  "fd_name_ch" : "自定义项1",
  "fd_name_en" : "selfDefine1",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "自定义项2",
  "fd_name_en" : "selfDefine2",
  "not_null" : true,
  "fd_value" : "aaa"
}],
"dev_version" : null,
"prod_version" : "1214597212252831744",
"dev_version_name" : null,
"prod_version_name" : "V1.1 [latest]",
"env_type" : "PROD_TYPE"
}]
}
}

```

**状态码：400**

BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

**状态码：401**

Unauthorized

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}

```

**状态码：403**

Forbidden

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

```

```
public class ListFactLogicTablesSolution {
    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListFactLogicTablesRequest request = new ListFactLogicTablesRequest();
        try {
            ListFactLogicTablesResponse response = client.listFactLogicTables(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```

request = ListFactLogicTablesRequest()
response = client.list_fact_logic_tables(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFactLogicTablesRequest{}
    response, err := client.ListFactLogicTables(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是FactLogicTableVO数组，以及总数。
400	BadRequest

状态码	描述
401	Unauthorized
403	Forbidden

## 8.18.2 删除事实表

### 功能介绍

根据ID集合删除事实表。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/fact-logic-tables

表 8-1104 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-1105 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

表 8-1106 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表, ID字符串。
del_types	否	String	删除类型。 枚举值: <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表, 只针对能物理化的表生效 (配置此值则表示删除)</li> </ul>

## 响应参数

状态码: 200

表 8-1107 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果, 返回成功删除的对象个数。

表 8-1108 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码: 400

表 8-1109 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1110 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1111 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数删除事实表。

```
DELETE https://{endpoint}/v2/{project_id}/design/fact-logic-tables
{
  "ids" : [ "1114492876374601728" ],
  "del_types" : "PHYSICAL_TABLE"
}
```

## 响应示例

状态码：200

Success，返回data数据是删除个数。



```
{
  "value": 1
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数删除事实表。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteDesignFactLogicTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
```

```
.withAk(ak)
.withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteDesignFactLogicTableRequest request = new DeleteDesignFactLogicTableRequest();
IdsParam body = new IdsParam();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1114492876374601728");
body.withDelTypes(IdsParam.DelTypeEnum.fromValue("PHYSICAL_TABLE"));
body.withIds(listbodyIds);
request.withBody(body);
try {
    DeleteDesignFactLogicTableResponse response = client.deleteDesignFactLogicTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据请求参数删除事实表。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteDesignFactLogicTableRequest()
        listbody = [
            "1114492876374601728"
        ]
        request.body = IdsParam(
            del_types="PHYSICAL_TABLE",
            ids=listbody
        )
```

```
response = client.delete_design_fact_logic_table(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据请求参数删除事实表。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDesignFactLogicTableRequest{
        var listIdsbody = []string{
            "1114492876374601728",
        }
        delTypesIdsParam:= model.GetIdsParamDelTypesEnum().PHYSICAL_TABLE
        request.Body = &model.IdsParam{
            DelTypes: &delTypesIdsParam,
            Ids: listIdsbody,
        }
    }
    response, err := client.DeleteDesignFactLogicTable(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

### 8.18.3 查看事实表详情

#### 功能介绍

通过ID查看事实表的详情信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/fact-logic-tables/{id}

表 8-1112 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID, ID字符串。

表 8-1113 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

## 请求参数

表 8-1114 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1115 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-1116 data

参数	参数类型	描述
value	<a href="#">FactLogicTableVO</a> object	value，统一的返回结果的外层数据结构。

表 8-1117 FactLogicTableVO

参数	参数类型	描述
id	String	事实表ID, ID字符串。
tb_name	String	表名称。
tb_logic_name	String	逻辑实体名。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象guid, ID字符串。
description	String	描述。
dw_name	String	数据连接名称, 只读, 创建和更新时无需填写。
owner	String	资产责任人。
create_by	String	创建人。
queue_name	String	dli数据连接执行sql所需的队列, 数据连接类型为DLI时必须。
dw_id	String	数据连接ID。
db_name	String	库名。
tb_id	String	数据表ID, 只读。
dim_table_ids	Array of strings	关联维度表ID。
reversed	Boolean	是否是逆向的。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
schema	String	DWS类型需要。

参数	参数类型	描述
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
table_type	String	表类型。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	String	表发布后，创建的数据目录技术资产guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产guid，只读，创建和更新时无需填写。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理, 只读。
dimensions	Array of <b>FactTableAttribut eVO</b> objects	事实表维度信息, 只读。
attributes	Array of <b>FactTableAttribut eVO</b> objects	事实表事实属性信息, 只读。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
measures	Array of <b>FactTableAttribut eVO</b> objects	事实表度量信息, 只读。
table_attributes	Array of <b>FactTableAttribut eVO</b> objects	事实表所有属性。
configs	String	其他配置
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
quality_id	String	质量ID, ID字符串。
alias	String	别名。

参数	参数类型	描述
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
obs_location	String	外表路径
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
model_id	Long	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1118 FactTableAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
fact_logic_table_id	String	所属事实表ID，只读，ID字符串。
ordinal	Integer	序号。
dimension_id	String	维度ID，ID字符串。
dimension_attr_id	String	维度属性ID，ID字符串。
role	String	维度角色。
dimension	<b>DimensionVO</b> object	dimension_id对应的维度信息，只读。
is_primary_key	Boolean	是否主键。

参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键，只读。
secrecy_levels	Array of <a href="#">SecrecyLevelRecordVO</a> objects	密级
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
name_en	String	英文名。
name_ch	String	中文名。
not_null	Boolean	是否不为空。

参数	参数类型	描述
attribute_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <a href="#">QualityInfoVO</a> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。

表 8-1119 DimensionVO

参数	参数类型	描述
id	String	编码（新建时只读，更新时必填），ID字符串。
name_en	String	维度英文名称。
dimension_type	String	维度类型（更新时只能由普通维度改为层级维度，其余场景都不允许修改）。 枚举值： <ul style="list-style-type: none"> <li>• COMMON: 普通维度</li> <li>• LOOKUP: 码表维度</li> <li>• HIERARCHIES: 层级维度</li> </ul>
name_ch	String	维度名称。

参数	参数类型	描述
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
code_table_id	String	引用码表ID（新建/更新“码表维度”时必须填），ID字符串。
code_table	<b>CodeTableVO</b> object	引用码表，只读。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，ID字符串。
hierarchies	Array of <b>DimensionHierar chiesVO</b> objects	层级属性。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag eVO</b> object	业务版本管理，只读。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
attributes	Array of <b>DimensionAttributeVO</b> objects	维度属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
datasource	<b>BizDatasourceRelationVO</b> object	数据源信息，新建/更新时必填。
owner	String	资产责任人，新建/更新时必填。
obs_location	String	外表路径
table_type	String	表类型。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
configs	String	其他配置
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。
prod_version_name	String	生产环境版本名称，只读。



参数	参数类型	描述
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	String	所属模型ID
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1120 CodeTableVO

参数	参数类型	描述
id	String	码表ID, ID字符串。
name_en	String	表名称, 英文名。
name_ch	String	表名称, 中文名。
tb_version	Integer	表版本。
directory_id	String	目录ID, ID字符串。
directory_path	String	目录树。
description	String	描述。
create_by	String	创建人。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>DRAFT: 草稿</li> <li>PUBLISH_DEVELOPING: 发布待审核</li> <li>PUBLISHED: 已发布</li> <li>OFFLINE_DEVELOPING: 下线待审核</li> <li>OFFLINE: 已下线</li> <li>REJECT: 已驳回</li> </ul>
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
code_table_fields	Array of <b>CodeTableFieldVO</b> objects	码表属性信息。

表 8-1121 CodeTableFieldVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
code_table_id	String	所属码表ID (更新时必填), ID字符串。
ordinal	Integer	序号。
name_en	String	字段名, 英文。
name_ch	String	字段名, 中文。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_unique_key	Boolean	是否唯一。
code_table_field_values	Array of <b>CodeTableFieldValueVO</b> objects	码表属性值。

参数	参数类型	描述
count_field_values	Integer	码表属性值总数。

表 8-1122 CodeTableFieldValueVO

参数	参数类型	描述
id	String	码表字段ID, ID字符串。
fd_id	String	所属码表属性ID, ID字符串。
fd_value	String	码表属性值。
ordinal	Integer	序号。
description	String	描述。

表 8-1123 DimensionHierarchiesVO

参数	参数类型	描述
id	String	编码, ID字符串。
name	String	层级名称。
attrs	Array of <a href="#">HierarchiesAttrVO</a> objects	层级包含的属性。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1124 HierarchiesAttrVO

参数	参数类型	描述
id	String	编码, ID字符串。
hierarchies_id	String	层级ID, ID字符串。

参数	参数类型	描述
attr_id	String	属性ID, ID字符串。
level	Integer	层次。
attr_name_en	String	引用属性编码。
attr_name_ch	String	引用属性名称, 只读。
detail_attr_ids	Array of strings	详情属性ID, ID字符串。
detail_attr_name_ens	Array of strings	详情属性英文。
detail_attr_name_chs	Array of strings	详情属性中文, 只读。
attr	<b>DimensionAttributeVO</b> object	引用属性字段, 只读。
detail_attrs	Array of <b>DimensionAttributeVO</b> objects	详情字段, 只读。

表 8-1125 DimensionAttributeVO

参数	参数类型	描述
id	String	编码, ID字符串。
dimension_id	String	维度ID, 只读, ID字符串。
code_table_field_id	String	码表属性ID, ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
create_by	String	创建人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值: <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>

参数	参数类型	描述
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。
is_biz_primary	Boolean	是否业务主键。
is_partition_key	Boolean	是否分区。
ordinal	Integer	序号。
not_null	Boolean	是否不为空。
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
alias	String	别名
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。

表 8-1126 ApprovalVO

参数	参数类型	描述
id	String	审批单ID, ID字符串。
tenant_id	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。



参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1127 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。

参数	参数类型	描述
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1128 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。
target_attr_id	String	目的字段ID，ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组，只读。
src_table_db_names	Array of strings	源表db名称数组，只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组，只读，ID字符串。
src_table_id_list	Array of strings	源表ID数组，只读，ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组，只读。
src_attr_id_list	Array of strings	源表字段ID数组，只读，ID字符串。
remark	String	备注。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-1129 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID，ID字符串。
table2_id	String	表2ID，ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1130 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID，ID字符串。
field2_id	String	属性2ID，ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-1131 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-1132 BizDatasourceRelationVO

参数	参数类型	描述
id	String	编码, ID字符串。
biz_id	String	业务对象信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_id	String	数据连接ID。
dw_name	String	数据连接名，只读。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。

表 8-1133 SecrecyLevelRecordVO

参数	参数类型	描述
id	String	属性关联密级的ID，ID字符串。
secrecyLevel_id	String	密级的ID，ID字符串。
secrecyLevel_name	String	密级名称。
uuid	String	数据安全主键。
slevel	Integer	密级等级。
description	String	密级描述。

参数	参数类型	描述
biz_id	String	业务对象ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建者。
update_by	String	更新者。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1134 QualityInfoVO

参数	参数类型	描述
id	String	编码ID，ID字符串。
table_id	String	表ID，只读，ID字符串。
attr_id	String	属性ID，只读，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1135 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-1136 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID, ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。

表 8-1137 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1138 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

### 状态码：400

表 8-1139 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

### 状态码：401

表 8-1140 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

### 状态码：403

表 8-1141 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

通过ID查看事实表的详情信息。

```
GET https://{endpoint}/v2/{project_id}/design/fact-logic-tables/1214584210761940992
```

## 响应示例

### 状态码：200

Success，返回data数据是FactLogicTableVO详情。

```
{  
  "data" : {
```



```

"value": {
  "id": "1214584210761940992",
  "tb_name": "fact_test_czh_0305_003",
  "tb_logic_name": "test_czh_0305_003",
  "l1_id": "1194966710038474752",
  "l2_id": null,
  "l3_id": null,
  "description": "无",
  "dw_name": "dws",
  "owner": "aaa",
  "create_by": "test_uesr",
  "queue_name": "default",
  "dw_id": "8259af52bd294f98920ebce75a426391",
  "db_name": "bi",
  "tb_id": "NativeTable-8259af52bd294f98920ebce75a426391-bi-fact_test_czh_0305_003",
  "dim_table_ids": null,
  "reversed": null,
  "partition_conf": null,
  "dirty_out_switch": false,
  "dirty_out_database": "",
  "dirty_out_prefix": "",
  "dirty_out_suffix": "",
  "schema": "",
  "distribute": null,
  "distribute_column": null,
  "obs_location": null,
  "table_type": "MANAGED",
  "status": "PUBLISHED",
  "tb_guid": null,
  "tb_logic_guid": "5417940a-0d1d-4fea-9b6f-65637da15112",
  "dw_type": "DWS",
  "l1": "test_czh_catalog",
  "l2": null,
  "l3": null,
  "create_time": "2024-03-05T14:44:02+08:00",
  "update_time": "2024-03-05T15:35:48+08:00",
  "approval_info": {
    "id": "1214597210327646208",
    "tenant_id": null,
    "name_ch": null,
    "name_en": null,
    "biz_id": "1214584210761940992",
    "biz_type": null,
    "biz_info": null,
    "biz_info_obj": null,
    "biz_version": 0,
    "biz_status": "PUBLISHED",
    "approval_status": "APPROVED",
    "approval_type": null,
    "submit_time": "2024-03-05T15:35:41+08:00",
    "create_by": null,
    "l1": null,
    "l2": null,
    "l3": null,
    "approval_time": "2024-03-05T15:35:41+08:00",
    "approver": "test_uesr",
    "email": null,
    "msg": "Fast Approval OK.",
    "directory_path": null
  },
  "new_biz": null,
  "dimensions": null,
  "attributes": null,
  "mappings": [],
  "measures": [ {
    "id": "1214584210866798592",
    "fact_logic_table_id": "1214584210761940992",
    "ordinal": 1,
    "dimension_id": null,

```

```

"role" : null,
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : null,
"secrecy_levels" : [ {
  "id" : "188",
  "secrecyLevel_id" : "1199020797063208960",
  "secrecyLevel_name" : "保密",
  "uuid" : "8a9481ba8c202835018c237ae6540007",
  "slevel" : 5,
  "description" : "答应我，一定不要对别人说哦。",
  "biz_id" : "1214584210866798592",
  "biz_type" : "FACT_LOGIC_TABLE",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-03-05T15:35:41+08:00",
  "update_time" : "2024-03-05T15:35:41+08:00"
}],
"description" : "",
"data_type" : "BIGINT",
"domain_type" : "NUMBER",
"data_type_extend" : "",
"name_en" : "aaa",
"name_ch" : "aaa",
"not_null" : false,
"attribute_type" : "FACT_MEASURE",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : [ ],
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : null
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : null
}
], {
  "id" : "1214584210866798593",
  "fact_logic_table_id" : "1214584210761940992",
  "ordinal" : 2,
  "dimension_id" : null,
  "role" : null,
  "dimension" : null,
  "is_primary_key" : false,
  "is_partition_key" : false,
  "is_foreign_key" : null,
  "secrecy_levels" : [ {
    "id" : "187",
    "secrecyLevel_id" : "1199020797063208960",
    "secrecyLevel_name" : "保密",
    "uuid" : "8a9481ba8c202835018c237ae6540007",
    "slevel" : 5,
    "description" : "答应我，一定不要对别人说哦。",
    "biz_id" : "1214584210866798593",
    "biz_type" : "FACT_LOGIC_TABLE",
    "create_by" : "test_uesr",
    "update_by" : "test_uesr",
    "create_time" : "2024-03-05T15:35:41+08:00",
    "update_time" : "2024-03-05T15:35:41+08:00"
  } ],
  "description" : "",
  "data_type" : "BIGINT",
  "domain_type" : "NUMBER",

```

```

"data_type_extend": "",
"name_en": "bbb",
"name_ch": "bbb",
"not_null": false,
"attribute_type": "FACT_MEASURE",
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": [],
"alias": "",
"self_defined_fields": [ {
  "fd_name_ch": "aa",
  "fd_name_en": "aa",
  "not_null": false,
  "fd_value": null
}, {
  "fd_name_ch": "bb",
  "fd_name_en": "bb",
  "not_null": false,
  "fd_value": null
} ]
}, {
  "id": "1214584210866798594",
  "fact_logic_table_id": "1214584210761940992",
  "ordinal": 3,
  "dimension_id": null,
  "role": null,
  "dimension": null,
  "is_primary_key": false,
  "is_partition_key": false,
  "is_foreign_key": null,
  "secrecy_levels": [ {
    "id": "186",
    "secrecyLevel_id": "1199020797063208960",
    "secrecyLevel_name": "保密",
    "uuid": "8a9481ba8c202835018c237ae6540007",
    "slevel": 5,
    "description": "答应我，一定不要对别人说哦。",
    "biz_id": "1214584210866798594",
    "biz_type": "FACT_LOGIC_TABLE",
    "create_by": "test_uesr",
    "update_by": "test_uesr",
    "create_time": "2024-03-05T15:35:41+08:00",
    "update_time": "2024-03-05T15:35:41+08:00"
  } ],
  "description": "",
  "data_type": "BIGINT",
  "domain_type": "NUMBER",
  "data_type_extend": "",
  "name_en": "ccc",
  "name_ch": "ccc",
  "not_null": false,
  "attribute_type": "FACT_MEASURE",
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": [],
  "alias": "",
  "self_defined_fields": [ {
    "fd_name_ch": "aa",
    "fd_name_en": "aa",
    "not_null": false,
    "fd_value": null
  }, {
    "fd_name_ch": "bb",
    "fd_name_en": "bb",
    "not_null": false,
    "fd_value": null
  } ]
} ],
"table_attributes": [ {

```

```

"id" : "1214584210866798592",
"fact_logic_table_id" : "1214584210761940992",
"ordinal" : 1,
"dimension_id" : null,
"role" : null,
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : null,
"secrecy_levels" : [ {
  "id" : "188",
  "secrecyLevel_id" : "1199020797063208960",
  "secrecyLevel_name" : "保密",
  "uuid" : "8a9481ba8c202835018c237ae6540007",
  "slevel" : 5,
  "description" : "答应我，一定不要对别人说哦。",
  "biz_id" : "1214584210866798592",
  "biz_type" : "FACT_LOGIC_TABLE",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-03-05T15:35:41+08:00",
  "update_time" : "2024-03-05T15:35:41+08:00"
} ],
"description" : "",
"data_type" : "BIGINT",
"domain_type" : "NUMBER",
"data_type_extend" : "",
"name_en" : "aaa",
"name_ch" : "aaa",
"not_null" : false,
"attribute_type" : "FACT_MEASURE",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : [ ],
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "aa",
  "fd_name_en" : "aa",
  "not_null" : false,
  "fd_value" : null
}, {
  "fd_name_ch" : "bb",
  "fd_name_en" : "bb",
  "not_null" : false,
  "fd_value" : null
} ]
}, {
"id" : "1214584210866798593",
"fact_logic_table_id" : "1214584210761940992",
"ordinal" : 2,
"dimension_id" : null,
"role" : null,
"dimension" : null,
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : null,
"secrecy_levels" : [ {
  "id" : "187",
  "secrecyLevel_id" : "1199020797063208960",
  "secrecyLevel_name" : "保密",
  "uuid" : "8a9481ba8c202835018c237ae6540007",
  "slevel" : 5,
  "description" : "答应我，一定不要对别人说哦。",
  "biz_id" : "1214584210866798593",
  "biz_type" : "FACT_LOGIC_TABLE",
  "create_by" : "test_uesr",
  "update_by" : "test_uesr",
  "create_time" : "2024-03-05T15:35:41+08:00",
  "update_time" : "2024-03-05T15:35:41+08:00"
} ]

```

```

    }],
    "description": "",
    "data_type": "BIGINT",
    "domain_type": "NUMBER",
    "data_type_extend": "",
    "name_en": "bbb",
    "name_ch": "bbb",
    "not_null": false,
    "attribute_type": "FACT_MEASURE",
    "stand_row_id": null,
    "stand_row_name": null,
    "quality_infos": [],
    "alias": "",
    "self_defined_fields": [ {
      "fd_name_ch": "aa",
      "fd_name_en": "aa",
      "not_null": false,
      "fd_value": null
    }, {
      "fd_name_ch": "bb",
      "fd_name_en": "bb",
      "not_null": false,
      "fd_value": null
    }
  ]
}, {
  "id": "1214584210866798594",
  "fact_logic_table_id": "1214584210761940992",
  "ordinal": 3,
  "dimension_id": null,
  "role": null,
  "dimension": null,
  "is_primary_key": false,
  "is_partition_key": false,
  "is_foreign_key": null,
  "secrecy_levels": [ {
    "id": "186",
    "secrecyLevel_id": "1199020797063208960",
    "secrecyLevel_name": "保密",
    "uuid": "8a9481ba8c202835018c237ae6540007",
    "slevel": 5,
    "description": "答应我，一定不要对别人说哦。",
    "biz_id": "1214584210866798594",
    "biz_type": "FACT_LOGIC_TABLE",
    "create_by": "test_uesr",
    "update_by": "test_uesr",
    "create_time": "2024-03-05T15:35:41+08:00",
    "update_time": "2024-03-05T15:35:41+08:00"
  }
  ],
  "description": "",
  "data_type": "BIGINT",
  "domain_type": "NUMBER",
  "data_type_extend": "",
  "name_en": "ccc",
  "name_ch": "ccc",
  "not_null": false,
  "attribute_type": "FACT_MEASURE",
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": [],
  "alias": "",
  "self_defined_fields": [ {
    "fd_name_ch": "aa",
    "fd_name_en": "aa",
    "not_null": false,
    "fd_value": null
  }, {
    "fd_name_ch": "bb",
    "fd_name_en": "bb",
    "not_null": false,

```

```
        "fd_value" : null
      }
    ]
  },
  "physical_table" : "UPDATE_SUCCESS",
  "dev_physical_table" : "NO_NEED",
  "technical_asset" : "CREATE_FAILED",
  "business_asset" : "UPDATE_SUCCESS",
  "meta_data_link" : "UPDATE_FAILED",
  "data_quality" : "NO_NEED",
  "summary_status" : "SUMMARY_FAILED",
  "quality_id" : null,
  "alias" : "",
  "configs" : "{}",
  "self_defined_fields" : [ {
    "fd_name_ch" : "自定义项1",
    "fd_name_en" : "selfDefine1",
    "not_null" : false,
    "fd_value" : ""
  }, {
    "fd_name_ch" : "自定义项2",
    "fd_name_en" : "selfDefine2",
    "not_null" : true,
    "fd_value" : "aaa"
  } ],
  "dev_version" : null,
  "prod_version" : "1214597212252831744",
  "dev_version_name" : null,
  "prod_version_name" : null,
  "env_type" : "PROD_TYPE"
}
}
```

#### 状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowFactLogicTableByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowFactLogicTableByIdRequest request = new ShowFactLogicTableByIdRequest();
        request.withId("{id}");
        try {
            ShowFactLogicTableByIdResponse response = client.showFactLogicTableById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowFactLogicTableByIdRequest()
    request.id = "{id}"
    response = client.show_fact_logic_table_by_id(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowFactLogicTableByIdRequest{}
    request.Id = "{id}"
    response, err := client.ShowFactLogicTableById(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。



## 状态码

状态码	描述
200	Success, 返回data数据是FactLogicTableVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.18.4 查看逆向事实表任务

### 功能介绍

查看逆向事实表任务。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/fact-logic-tables/database

表 8-1142 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-1143 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 (响应消息头中X-Subject-Token的值) 使用Token认证时必选。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1144 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-1145 data

参数	参数类型	描述
value	<a href="#">BatchOperationVO</a> object	value，统一的返回结果的外层数据结构。

表 8-1146 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID，ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID，ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 运行中</li> <li>• SUCCESS: 操作成功</li> <li>• FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。
groups	Array of <b>BatchOperationVO</b> objects	分组信息。

**状态码：400**

表 8-1147 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

**状态码：401**

表 8-1148 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

**状态码：403**

表 8-1149 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

查看逆向事实表任务。

GET https://{endpoint}/v2/{project\_id}/design/fact-logic-tables/database

## 响应示例

状态码: 200

Success

```
{
  "data": {
    "value": {
      "id": null,
      "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70",
      "group_id": "1231900573092171776",
      "biz_name": null,
      "biz_id": null,
      "operation_status": "RUNNING",
      "operation_type": "FACT_LOGIC_TABLE_REVERSE_DB",
      "biz_info": null,
      "create_by": "test_uesr",
      "remark": null,
      "total": 0,
      "success": 0,
      "failed": 0,
      "rate": null,
      "logs": null,
      "groups": [ ]
    }
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CheckFactLogicTableStatusSolution {
    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
CheckFactLogicTableStatusRequest request = new CheckFactLogicTableStatusRequest();
try {
    CheckFactLogicTableStatusResponse response = client.checkFactLogicTableStatus(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CheckFactLogicTableStatusRequest()
        response = client.check_fact_logic_table_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CheckFactLogicTableStatusRequest{}
    response, err := client.CheckFactLogicTableStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.19 汇总表接口

### 8.19.1 查找汇总表

#### 功能介绍

通过中英文名称、创建者、审核人、状态、修改时间分页查找汇总表信息，中英文名称支持模糊查询。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/aggregation-logic-tables

表 8-1150 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1151 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
name_ch	否	String	按名称精确查询。
name_en	否	String	按编码精确查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。
owner	否	String	按负责人查询。

参数	是否必选	参数类型	描述
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
sync_status	否	String	同步状态枚举。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 同步中</li> <li>• NO_NEED: 未同步</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> </ul>
sync_key	否	Array of strings	同步任务类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• BUSINESS_ASSET: 同步业务资产</li> <li>• DATA_QUALITY: 创建质量作业</li> <li>• TECHNICAL_ASSET: 同步技术资产</li> <li>• META_DATA_LINK: 资产关联</li> <li>• PHYSICAL_TABLE: 创建表（生产环境）</li> <li>• DEV_PHYSICAL_TABLE: 创建表（开发环境）</li> <li>• DLF_TASK: 创建数据开发作业</li> <li>• MATERIALIZATION: 数值落库（码表）</li> <li>• PUBLISH_TO_DLM: 发布数据服务API</li> <li>• SUMMARY_STATUS: 整体状态</li> </ul>



参数	是否必选	参数类型	描述
l3_id	否	String	业务对象l3的ID，ID字符串。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。
biz_catalog_id	否	String	所属的业务分层的ID。
auto_generate	否	Boolean	是否自动生成

## 请求参数

表 8-1152 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-1153 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回数据。

表 8-1154 data

参数	参数类型	描述
value	<b>value</b> object	返回的数据信息。

表 8-1155 value

参数	参数类型	描述
records	Array of <b>AggregationLogicTableVO</b> objects	查询到的汇总表值对象 ( AggregationLogicTableVO ) 数组。
total	Integer	符合搜索条件的记录总数。

表 8-1156 AggregationLogicTableVO

参数	参数类型	描述
id	String	汇总表的唯一系统ID, 更新时必须填, 创建时不须填写, ID字符串。
tb_name	String	汇总表英文名称, 对应实际的物理表名。
tb_logic_name	String	汇总表的中文名, 用于展示使用。

参数	参数类型	描述
l1_id	String	主题域分组ID，只读，创建和更新时无需填写，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	汇总表所属主题的ID，必填，ID字符串。
description	String	汇总表描述信息。
owner	String	汇总表的资产责任人。
secret_type	String	密级类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLIC: 公开</li> <li>• SECRET: 秘密</li> <li>• CONFIDENTIAL: 机密</li> <li>• SUPER_SECRET: 绝密</li> </ul>
apply_bg	String	适用bg。
create_by	String	汇总表的创建人，只读，创建和更新时无需填写。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须填写。
dw_id	String	汇总表所在的数据连接ID，为32位十六进制数字。
db_name	String	汇总表所在的数据库名。
tb_id	String	汇总表创建的表ID，是服务内部ID，只读，创建和更新时无需填写
schema	String	汇总表所在的Schema名，DWS类型必须填写。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	String	表发布后，创建的数据目录技术资产 guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产 guid，只读，创建和更新时无需填写。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
create_time	String	创建时间，只读，创建和更新时无需填写。
update_time	String	更新时间，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。
dimension_group	String	颗粒度ID。
group_name	String	颗粒度名称，只读。
group_code	String	颗粒度编码，只读。

参数	参数类型	描述
time_period	<b>AggregationLogi cTableAttributeV O</b> object	汇总表时间周期属性信息。
table_attributes	Array of <b>AggregationLogi cTableAttributeV O</b> objects	汇总表属性信息，依据attribute_type判断类型。
physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
compression	String	DWS数据压缩等级，列压缩等级为no/low/middle/high，行压缩等级为no/yes。 枚举值： <ul style="list-style-type: none"> <li>• "NO": 不压缩</li> <li>• "YES": 压缩</li> <li>• "LOW": 低等级压缩</li> <li>• "MIDDLE": 中等级压缩</li> <li>• "HIGH": 高等级压缩</li> </ul>
obs_location	String	外表路径。
pre_combine_field	String	版本字段。
table_type	String	表类型。
dlf_task_id	String	DLF作业ID。
quality_id	String	质量ID，ID字符串。
reversed	Boolean	是否是逆向的，只读。



参数	参数类型	描述
table_version	Integer	为2时，表示汇总表是汇总生成的，只读。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
alias	String	别名。
configs	String	其他配置。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
api_id	String	API ID。
sql	String	汇总表绑定的SQL。
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>• INVALID_TYPE: 无效环境</li> <li>• DEV_TYPE: 开发环境</li> <li>• PROD_TYPE: 生产环境</li> <li>• DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	Long	所属模型ID。
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1157 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1158 AggregationLogicTableAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
aggregation_logic_table_id	String	所属汇总表ID，ID字符串。
ordinal	Integer	序号。
name_en	String	字段名。
name_ch	String	业务属性。

参数	参数类型	描述
attribute_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
is_primary_key	Boolean	是否主键。
is_partition_key	Boolean	是否分区键。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级
not_null	Boolean	是否不为空。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
ref_id	String	属性关联对象的id
ref_name_ch	String	属性关联对象的中文名
ref_name_en	String	属性关联对象的英文名
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。

参数	参数类型	描述
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。

表 8-1159 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1160 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1161 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1162 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-1163 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1164 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1165 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1166 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

查询手动创建、模糊查询中文名含有 ‘test\_d’ 的前十条汇总表数据

```
GET https://{endpoint}/v2/{project_id}/design/aggregation-logic-tables?
offset=1&limit=10&name=test_d&auto_generate=false
```

## 响应示例

**状态码：200**

Success, 返回data数据是AggregationLogicTableVO数组, 以及总数。

```
{
  "data": {
    "value": {
      "total": "1",
      "records": [ {
        "id": "874242075850342400",
        "tb_name": "test_dws_rate_code",
        "tb_logic_name": "费率代码汇总",
        "l1_id": "873232498044411904",
        "l2_id": "873232716601204736",
        "l3_id": "873233519286059008",
        "description": "无",
        "owner": "abc",
        "create_by": "abc",
        "queue_name": "dgc_dws_test",
        "dw_id": "2c9080f07b03db8d017b198a4a20002f",
        "db_name": "demo_dm_db",
        "tb_id": "NativeTable-2c9080f07b03db8d017b198a4a20002f-demo_dm_db-dws_rate_code",
        "schema": "",
        "dw_name": null,
        "status": "PUBLISHED",
        "tb_guid": "a268644b-31b6-42e5-84ec-e35c31b541a5",
        "tb_logic_guid": "8948f715-3567-4684-9950-e3ed01e77121",
        "dw_type": "DWS",
        "l1": "城市交通",
        "l2": "行程记录",
        "l3": "记录统计",
        "create_time": "2021-08-09T10:45:54+08:00",
        "update_time": "2021-09-26T15:36:53+08:00",
        "approval_info": {
          "id": "874242076399796224",
          "tenant_id": null,
          "name_ch": null,
          "name_en": null,
          "biz_id": "874242075850342400",
          "biz_type": null,
          "biz_info": null,
          "biz_info_obj": null,
          "biz_version": 0,
          "biz_status": "PUBLISHED",
          "approval_status": "APPROVED",
          "approval_type": null,
          "submit_time": "2021-08-09T10:45:54+08:00",
          "create_by": null,
          "l1": null,
          "l2": null,
          "l3": null,
        }
      }
    ]
  }
}
```



```

"approval_time": "2021-08-09T10:45:53+08:00",
"approver": "abc",
"email": null,
"msg": "Fast Approval OK.",
"directory_path": null
},
"new_biz": null,
"dimension_group": "873252544997130240",
"group_name": "费率代码",
"group_code": null,
"time_period": {
  "id": "874242075984560130",
  "aggregation_logic_table_id": "874242075850342400",
  "ordinal": 1,
  "name_en": "dtime",
  "name_ch": "统计日期",
  "attribute_type": "SUMMARY_TIME",
  "is_primary_key": false,
  "is_partition_key": true,
  "not_null": false,
  "description": null,
  "data_type": "TIMESTAMP",
  "domain_type": "DATETIME",
  "data_type_extend": null,
  "ref_id": null,
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": null,
  "alias": null
},
"table_attributes": [ {
  "id": "874242075984560130",
  "aggregation_logic_table_id": "874242075850342400",
  "ordinal": 1,
  "name_en": "dtime",
  "name_ch": "统计日期",
  "attribute_type": "SUMMARY_TIME",
  "is_primary_key": false,
  "is_partition_key": true,
  "not_null": false,
  "description": null,
  "data_type": "TIMESTAMP",
  "domain_type": "DATETIME",
  "data_type_extend": null,
  "ref_id": null,
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": null,
  "alias": null
}, {
  "id": "874242075984560131",
  "aggregation_logic_table_id": "874242075850342400",
  "ordinal": 2,
  "name_en": "sum_total_amount",
  "name_ch": "总车费(费率代码)",
  "attribute_type": "DERIVATIVE_INDEX",
  "is_primary_key": false,
  "is_partition_key": false,
  "not_null": false,
  "description": "",
  "data_type": "DOUBLE",
  "domain_type": "NUMBER",
  "data_type_extend": null,
  "ref_id": "873265784540004352",
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": null,
  "alias": null
}, {

```

```

    "id" : "874242075984560132",
    "aggregation_logic_table_id" : "874242075850342400",
    "ordinal" : 3,
    "name_en" : "dim_rate_code_rate_code_id",
    "name_ch" : "费率代码_费率id",
    "attribute_type" : "SUMMARY_DIMENSION_ATTRIBUTE",
    "is_primary_key" : false,
    "is_partition_key" : false,
    "not_null" : true,
    "description" : "",
    "data_type" : "BIGINT",
    "domain_type" : "NUMBER",
    "data_type_extend" : null,
    "ref_id" : "873252544997130240.873258130849636352",
    "stand_row_id" : null,
    "stand_row_name" : null,
    "quality_infos" : null,
    "alias" : null
  }, {
    "id" : "874242075984560133",
    "aggregation_logic_table_id" : "874242075850342400",
    "ordinal" : 4,
    "name_en" : "dim_rate_code_rate_code_value",
    "name_ch" : "费率代码_费率说明",
    "attribute_type" : "SUMMARY_DIMENSION_ATTRIBUTE",
    "is_primary_key" : false,
    "is_partition_key" : false,
    "not_null" : false,
    "description" : "",
    "data_type" : "STRING",
    "domain_type" : "STRING",
    "data_type_extend" : null,
    "ref_id" : "873252544997130240.873258130849636353",
    "stand_row_id" : null,
    "stand_row_name" : null,
    "quality_infos" : null,
    "alias" : null
  } ],
  "physical_table" : "CREATE_SUCCESS",
  "technical_asset" : "UPDATE_SUCCESS",
  "business_asset" : "UPDATE_SUCCESS",
  "meta_data_link" : "UPDATE_SUCCESS",
  "data_quality" : "UPDATE_SUCCESS",
  "dlf_task" : "CREATE_SUCCESS",
  "publish_to_dlm" : "NO_NEED",
  "summary_status" : "SUMMARY_SUCCESS",
  "distribute" : null,
  "distribute_column" : null,
  "table_type" : "MANAGED",
  "dlf_task_id" : "demo_dm_db_dws_rate_code_874242075850342400",
  "quality_id" : "874242188178075649",
  "reversed" : null,
  "table_version" : 0,
  "partition_conf" : null,
  "dirty_out_switch" : false,
  "dirty_out_database" : "",
  "dirty_out_prefix" : "",
  "dirty_out_suffix" : "",
  "alias" : "",
  "self_defined_fields" : [ ],
  "api_id" : null
} ]
}
}
}

```

**状态码: 400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListAggregationLogicTablesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAggregationLogicTablesRequest request = new ListAggregationLogicTablesRequest();
        try {
            ListAggregationLogicTablesResponse response = client.listAggregationLogicTables(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAggregationLogicTablesRequest()
        response = client.list_aggregation_logic_tables(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```

projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build())

request := &model.ListAggregationLogicTablesRequest{}
response, err := client.ListAggregationLogicTables(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是AggregationLogicTableVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.19.2 新建汇总表

### 功能介绍

根据入参，手动创建汇总表。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/aggregation-logic-tables

表 8-1167 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-1168 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1169 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	汇总表的唯一系统ID，更新时必填，创建时不须填写，ID字符串。
tb_name	是	String	汇总表英文名称，对应实际的物理表名。
tb_logic_name	是	String	汇总表的中文名，用于展示使用。
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。

参数	是否必选	参数类型	描述
l3_id	是	String	汇总表所属主题的ID，必填，ID字符串。
description	否	String	汇总表描述信息。
owner	是	String	汇总表的资产责任人。
secret_type	否	String	密级类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLIC: 公开</li> <li>• SECRET: 秘密</li> <li>• CONFIDENTIAL: 机密</li> <li>• SUPER_SECRET: 绝密</li> </ul>
apply_bg	否	String	适用bg。
queue_name	否	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须填写。
dw_id	是	String	汇总表所在的数据连接ID，为32位十六进制数字。
db_name	是	String	汇总表所在的数据库名。
schema	否	String	汇总表所在的Schema名，DWS类型必须填写。
dw_name	否	String	数据连接名称，只读，创建和更新时无需填写。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	否	String	表发布后，创建的数据目录技术资产guid，只读，创建和更新时无需填写。
tb_logic_guid	否	String	表发布后，创建的数据目录业务资产guid，只读，创建和更新时无需填写。

参数	是否必选	参数类型	描述
dw_type	是	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManageVO object	业务版本管理，只读。
dimension_group	否	String	颗粒度ID。
time_period	否	AggregationLogicTableAttributeVO object	汇总表时间周期属性信息。
table_attributes	否	Array of AggregationLogicTableAttributeVO objects	汇总表属性信息，依据 attribute_type判断类型。
distribute	否	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行 Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	否	String	DISTRIBUTE BY HASH column.



参数	是否必选	参数类型	描述
compression	否	String	DWS数据压缩等级，列压缩等级为no/low/middle/high，行压缩等级为no/yes。 枚举值： <ul style="list-style-type: none"> <li>• "NO": 不压缩</li> <li>• "YES": 压缩</li> <li>• "LOW": 低等级压缩</li> <li>• "MIDDLE": 中等级压缩</li> <li>• "HIGH": 高等级压缩</li> </ul>
obs_location	否	String	外表路径。
pre_combine_field	否	String	版本字段。
table_type	否	String	表类型。
dlf_task_id	否	String	DLF作业ID。
quality_id	否	String	质量ID，ID字符串。
partition_conf	否	String	分区表达式。
dirty_out_switch	否	Boolean	异常数据输出开关。
dirty_out_database	否	String	异常数据输出库。
dirty_out_prefix	否	String	异常表前缀。
dirty_out_suffix	否	String	异常表后缀。
alias	否	String	别名。
configs	否	String	其他配置。
self_defined_fields	否	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。
api_id	否	String	API ID。
sql	否	String	汇总表绑定的SQL。
model_id	否	Long	所属模型ID。
model	否	<a href="#">WorkspaceVO</a> object	模型信息。

表 8-1170 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID, ID字符串。
tenant_id	否	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-1171 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。



参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-1172 AggregationLogicTableAttributeVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
aggregation_logic_table_id	否	String	所属汇总表ID，ID字符串。
ordinal	否	Integer	序号。
name_en	是	String	字段名。
name_ch	是	String	业务属性。

参数	是否必选	参数类型	描述
attribute_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
is_primary_key	是	Boolean	是否主键。
is_partition_key	是	Boolean	是否分区键。
not_null	否	Boolean	是否不为空。
description	否	String	描述。
data_type	是	String	字段类型。
data_type_extend	否	String	数据类型扩展字段。
ref_id	否	String	属性关联对象的id

参数	是否必选	参数类型	描述
stand_row_id	否	String	关联的数据标准的ID, ID字符串。
alias	否	String	别名。

表 8-1173 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

表 8-1174 WorkspaceVO

参数	是否必选	参数类型	描述
id	否	String	编号, ID字符串。
name	是	String	工作区名字。
description	否	String	描述。
is_physical	否	Boolean	是否为物理表。
frequent	否	Boolean	是否为常用。
top	否	Boolean	分层治理。
level	否	String	数据治理分层。 枚举值: <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	否	String	数据连接类型, 对应表所在的数仓类型, 取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_by	否	String	创建人。
update_by	否	String	更新人。

参数	是否必选	参数类型	描述
type	是	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	否	String	关联的业务分层的ID列表 { "l1Ids": [], "l2Ids": [], "l3Ids": [] }。
databases	否	Array of strings	数据库名称数组。
table_model_prefix	否	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

## 响应参数

状态码：200

表 8-1175 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回数据。

表 8-1176 data

参数	参数类型	描述
value	<b>AggregationLogicTableVO</b> object	返回的数据信息。

表 8-1177 AggregationLogicTableVO

参数	参数类型	描述
id	String	汇总表的唯一系统ID，更新时必填，创建时不须填写，ID字符串。
tb_name	String	汇总表英文名称，对应实际的物理表名。
tb_logic_name	String	汇总表的中文名，用于展示使用。

参数	参数类型	描述
l1_id	String	主题域分组ID，只读，创建和更新时无需填写，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	汇总表所属主题的ID，必填，ID字符串。
description	String	汇总表描述信息。
owner	String	汇总表的资产责任人。
secret_type	String	密级类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLIC: 公开</li> <li>• SECRET: 秘密</li> <li>• CONFIDENTIAL: 机密</li> <li>• SUPER_SECRET: 绝密</li> </ul>
apply_bg	String	适用bg。
create_by	String	汇总表的创建人，只读，创建和更新时无需填写。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须填写。
dw_id	String	汇总表所在的数据连接ID，为32位十六进制数字。
db_name	String	汇总表所在的数据库名。
tb_id	String	汇总表创建的表ID，是服务内部ID，只读，创建和更新时无需填写
schema	String	汇总表所在的Schema名，DWS类型必须填写。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	String	表发布后，创建的数据目录技术资产 guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产 guid，只读，创建和更新时无需填写。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
create_time	String	创建时间，只读，创建和更新时无需填写。
update_time	String	更新时间，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。
dimension_group	String	颗粒度ID。
group_name	String	颗粒度名称，只读。
group_code	String	颗粒度编码，只读。

参数	参数类型	描述
time_period	<b>AggregationLogi cTableAttributeV O</b> object	汇总表时间周期属性信息。
table_attributes	Array of <b>AggregationLogi cTableAttributeV O</b> objects	汇总表属性信息，依据attribute_type判断类型。
physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>NO_NEED: 未同步</li> <li>CREATE_SUCCESS: 创建成功</li> <li>CREATE_FAILED: 创建失败</li> <li>UPDATE_SUCCESS: 更新成功</li> <li>UPDATE_FAILED: 更新失败</li> <li>SUMMARY_SUCCESS: 整体成功</li> <li>SUMMARY_FAILED: 整体失败</li> <li>RUNNING: 整体运行</li> <li>OFFLINE: 已下线</li> </ul>
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
compression	String	DWS数据压缩等级，列压缩等级为no/low/middle/high，行压缩等级为no/yes。 枚举值： <ul style="list-style-type: none"> <li>"NO": 不压缩</li> <li>"YES": 压缩</li> <li>"LOW": 低等级压缩</li> <li>"MIDDLE": 中等级压缩</li> <li>"HIGH": 高等级压缩</li> </ul>
obs_location	String	外表路径。
pre_combine_field	String	版本字段。
table_type	String	表类型。
dlf_task_id	String	DLF作业ID。
quality_id	String	质量ID，ID字符串。
reversed	Boolean	是否是逆向的，只读。

参数	参数类型	描述
table_version	Integer	为2时，表示汇总表是汇总生成的，只读。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
alias	String	别名。
configs	String	其他配置。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
api_id	String	API ID。
sql	String	汇总表绑定的SQL。
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	Long	所属模型ID。
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1178 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1179 AggregationLogicTableAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
aggregation_logic_table_id	String	所属汇总表ID，ID字符串。
ordinal	Integer	序号。
name_en	String	字段名。
name_ch	String	业务属性。



参数	参数类型	描述
attribute_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
is_primary_key	Boolean	是否主键。
is_partition_key	Boolean	是否分区键。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级
not_null	Boolean	是否不为空。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
ref_id	String	属性关联对象的id
ref_name_ch	String	属性关联对象的中文名
ref_name_en	String	属性关联对象的英文名
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。

参数	参数类型	描述
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。

表 8-1180 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1181 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1182 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-1183 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-1184 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1185 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1186 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1187 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

根据入参，手动创建汇总表。

POST https://{endpoint}/v2/{project\_id}/design/aggregation-logic-tables

```
{
  "tb_name": "test_create_aggregation_logic_table",
  "tb_logic_name": "test_create_aggregation_logic_table",
  "alias": "",
  "dimension_group": "0",
  "secret_type": "PUBLIC",
  "apply_bg": "",
  "l3_id": "1105525024175058944",
  "description": "无",
  "owner": "autotest_tics",
  "dw_type": "DWS",
  "dw_id": "8259af52bd294f98920ebce75a426391",
  "db_name": "bi",
  "schema": "",
  "queue_name": "default",
  "self_defined_fields": [ {
    "fd_name_ch": "aaa",
    "fd_name_en": "aaa",
    "not_null": false,
    "fd_value": ""
  } ],
  "configs": "{}",
  "table_type": "MANAGED",
  "obs_location": "",
  "table_attributes": [ {
    "id": null,
    "name_ch": "col_aa",
    "name_en": "col_aa",
    "ordinal": 1,
    "is_primary_key": false,
    "is_partition_key": false,
    "not_null": false,
    "description": "",
    "data_type": "BIGINT",
    "data_type_extend": "",
    "stand_row_id": null,
    "stand_row_name": null,
    "ref_id": null,
    "ref_name_ch": null,
    "secrecy_levels": [ {
      "name": "secrecy_level_name"
    } ]
  } ],
  "id": 0,
  "sql": ""
}
```

## 响应示例

**状态码：200**

Success, 返回data数据是AggregationLogicTableVO

```
{
  "data": {
```

```

"value": {
  "id": "1227937502401331200",
  "tb_name": "test_create_aggregation_logic_table",
  "tb_logic_name": "test_create_aggregation_logic_table",
  "l1_id": null,
  "l2_id": null,
  "l3_id": "1105525024175058944",
  "description": "无",
  "owner": "autotest_tics",
  "create_by": "test_uesr",
  "queue_name": "default",
  "dw_id": "8259af52bd294f98920ebce75a426391",
  "db_name": "bi",
  "tb_id": null,
  "schema": "",
  "dw_name": null,
  "status": "DRAFT",
  "tb_guid": null,
  "tb_logic_guid": null,
  "dw_type": "DWS",
  "l1": null,
  "l2": null,
  "l3": null,
  "create_time": "2024-04-11T11:05:14.548+08:00",
  "update_time": "2024-04-11T11:05:14.548+08:00",
  "approval_info": null,
  "new_biz": null,
  "dimension_group": "0",
  "group_name": null,
  "group_code": null,
  "time_period": null,
  "table_attributes": [ {
    "id": null,
    "aggregation_logic_table_id": null,
    "ordinal": 1,
    "name_en": "col_aa",
    "name_ch": "col_aa",
    "attribute_type": null,
    "is_primary_key": false,
    "is_partition_key": false,
    "not_null": false,
    "secrecy_levels": null,
    "description": "",
    "data_type": "BIGINT",
    "domain_type": "NUMBER",
    "data_type_extend": "",
    "ref_id": null,
    "ref_name_ch": null,
    "ref_name_en": null,
    "stand_row_id": null,
    "stand_row_name": null,
    "quality_infos": null,
    "alias": null
  }, {
    "id": null,
    "aggregation_logic_table_id": null,
    "ordinal": 2,
    "name_en": "col_bb",
    "name_ch": "col_bb",
    "attribute_type": null,
    "is_primary_key": false,
    "is_partition_key": false,
    "not_null": false,
    "secrecy_levels": null,
    "description": "",
    "data_type": "BIGINT",
    "domain_type": "NUMBER",
    "data_type_extend": "",
    "ref_id": null,
  }
}

```

```
"ref_name_ch" : null,
"ref_name_en" : null,
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : null
}],
"physical_table" : null,
"dev_physical_table" : null,
"technical_asset" : null,
"business_asset" : null,
"meta_data_link" : null,
"data_quality" : null,
"dlf_task" : null,
"publish_to_dlm" : null,
"summary_status" : null,
"distribute" : null,
"distribute_column" : null,
"compression" : null,
"obs_location" : null,
"pre_combine_field" : null,
"table_type" : "MANAGED",
"dlf_task_id" : null,
"quality_id" : null,
"reversed" : false,
"table_version" : 0,
"partition_conf" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"alias" : "",
"configs" : "{}",
"self_defined_fields" : [ {
  "fd_name_ch" : "aaa",
  "fd_name_en" : "aaa",
  "not_null" : false,
  "fd_value" : ""
}],
"api_id" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : null,
"sql" : ""
}
}
```

#### 状态码：400

##### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

##### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

## Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据入参，手动创建汇总表。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateDesignAggregationLogicTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateDesignAggregationLogicTableRequest request = new
        CreateDesignAggregationLogicTableRequest();
        AggregationLogicTableVO body = new AggregationLogicTableVO();
        List<SelfDefinedFieldVO> listbodySelfDefinedFields = new ArrayList<>();
        listbodySelfDefinedFields.add(
            new SelfDefinedFieldVO()
                .withFdNameCh("aaa")
                .withFdNameEn("aaa")
                .withNotNull(false)
                .withFdValue("")
        );
        List<SecrecyLevelVO> listTableAttributesSecrecyLevels = new ArrayList<>();
        listTableAttributesSecrecyLevels.add(
            new SecrecyLevelVO()
                .withName("secrecy_level_name")
        );
        List<AggregationLogicTableAttributeVO> listbodyTableAttributes = new ArrayList<>();
```

```
listbodyTableAttributes.add(
    new AggregationLogicTableAttributeVO()
        .withOrdinal(1)
        .withNameEn("col_aa")
        .withNameCh("col_aa")
        .withIsPrimaryKey(false)
        .withIsPartitionKey(false)
        .withSecrecyLevels(listTableAttributesSecrecyLevels)
        .withNotNull(false)
        .withDescription("")
        .withDataType("BIGINT")
        .withDataTypeExtend("")
    );
body.withSql("");
body.withSelfDefinedFields(listbodySelfDefinedFields);
body.withConfigs("{}");
body.withAlias("");
body.withTableType("MANAGED");
body.withObsLocation("");
body.withTableAttributes(listbodyTableAttributes);
body.withDimensionGroup("0");
body.withDwType("DWS");
body.withSchema("");
body.withDbName("bi");
body.withDwId("8259af52bd294f98920ebce75a426391");
body.withQueueName("default");
body.withApplyBg(AggregationLogicTableVO.ApplyBgEnum.fromValue(""));
body.withSecretType(AggregationLogicTableVO.SecretTypeEnum.fromValue("PUBLIC"));
body.withOwner("autotest_tics");
body.withDescription("无");
body.withL3Id("1105525024175058944");
body.withTbLogicName("test_create_aggregation_logic_table");
body.withTbName("test_create_aggregation_logic_table");
body.withId("0");
request.withBody(body);
try {
    CreateDesignAggregationLogicTableResponse response =
client.createDesignAggregationLogicTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据入参，手动创建汇总表。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateDesignAggregationLogicTableRequest()
    listSelfDefinedFieldsbody = [
        SelfDefinedFieldVO(
            fd_name_ch="aaa",
            fd_name_en="aaa",
            not_null=False,
            fd_value=""
        )
    ]
    listSecrecyLevelsTableAttributes = [
        SecrecyLevelVO(
            name="secrecy_level_name"
        )
    ]
    listTableAttributesbody = [
        AggregationLogicTableAttributeVO(
            ordinal=1,
            name_en="col_aa",
            name_ch="col_aa",
            is_primary_key=False,
            is_partition_key=False,
            secrecy_levels=listSecrecyLevelsTableAttributes,
            not_null=False,
            description="",
            data_type="BIGINT",
            data_type_extend=""
        )
    ]
    request.body = AggregationLogicTableVO(
        sql="",
        self_defined_fields=listSelfDefinedFieldsbody,
        configs="{ }",
        alias="",
        table_type="MANAGED",
        obs_location="",
        table_attributes=listTableAttributesbody,
        dimension_group="0",
        dw_type="DWS",
        schema="",
        db_name="bi",
        dw_id="8259af52bd294f98920ebce75a426391",
        queue_name="default",
        apply_bg="",
        secret_type="PUBLIC",
        owner="autotest_tics",
        description="无",
        l3_id="1105525024175058944",
        tb_logic_name="test_create_aggregation_logic_table",
        tb_name="test_create_aggregation_logic_table",
        id="0"
    )
    response = client.create_design_aggregation_logic_table(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
```



```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

根据入参，手动创建汇总表。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDesignAggregationLogicTableRequest{
        fdNameChSelfDefinedFields:= "aaa"
        fdNameEnSelfDefinedFields:= "aaa"
        notNullSelfDefinedFields:= false
        fdValueSelfDefinedFields:= ""
        var listSelfDefinedFieldsbody = []model.SelfDefinedFieldVo{
            {
                FdNameCh: &fdNameChSelfDefinedFields,
                FdNameEn: &fdNameEnSelfDefinedFields,
                NotNull: &notNullSelfDefinedFields,
                FdValue: &fdValueSelfDefinedFields,
            },
        }
        var listSecrecyLevelsTableAttributes = []model.SecrecyLevelVo{
            {
                Name: "secrecy_level_name",
            },
        }
        ordinalTableAttributes:= int32(1)
        notNullTableAttributes:= false
        descriptionTableAttributes:= ""
        dataTypeExtendTableAttributes:= ""
        var listTableAttributesbody = []model.AggregationLogicTableAttributeVo{
            {
                Ordinal: &ordinalTableAttributes,
                NameEn: "col_aa",
                NameCh: "col_aa",
                IsPrimaryKey: false,
                IsPartitionKey: false,
```

```

        SecrecyLevels: &listSecrecyLevelsTableAttributes,
        NotNull: &notNullTableAttributes,
        Description: &descriptionTableAttributes,
        DataType: "BIGINT",
        DataTypeExtend: &dataTypeExtendTableAttributes,
    },
}
sqlAggregationLogicTableVo:= ""
configsAggregationLogicTableVo:= "{}"
aliasAggregationLogicTableVo:= ""
tableTypeAggregationLogicTableVo:= "MANAGED"
obsLocationAggregationLogicTableVo:= ""
dimensionGroupAggregationLogicTableVo:= "0"
schemaAggregationLogicTableVo:= ""
queueNameAggregationLogicTableVo:= "default"
applyBgApplyBg:= model.GetApplyBgEnumApplyBgEnum().EMPTY
secretTypeSecretType:= model.GetSecretTypeEnumSecretTypeEnum().PUBLIC
descriptionAggregationLogicTableVo:= "无"
idAggregationLogicTableVo:= "0"
request.Body = &model.AggregationLogicTableVo{
    Sql: &sqlAggregationLogicTableVo,
    SelfDefinedFields: &listSelfDefinedFieldsbody,
    Configs: &configsAggregationLogicTableVo,
    Alias: &aliasAggregationLogicTableVo,
    TableType: &tableTypeAggregationLogicTableVo,
    ObsLocation: &obsLocationAggregationLogicTableVo,
    TableAttributes: &listTableAttributesbody,
    DimensionGroup: &dimensionGroupAggregationLogicTableVo,
    DwType: "DWS",
    Schema: &schemaAggregationLogicTableVo,
    DbName: "bi",
    DwId: "8259af52bd294f98920ebce75a426391",
    QueueName: &queueNameAggregationLogicTableVo,
    ApplyBg: &applyBgApplyBg,
    SecretType: &secretTypeSecretType,
    Owner: "autotest_tics",
    Description: &descriptionAggregationLogicTableVo,
    L3Id: "1105525024175058944",
    TbLogicName: "test_create_aggregation_logic_table",
    TbName: "test_create_aggregation_logic_table",
    Id: &idAggregationLogicTableVo,
}
response, err := client.CreateDesignAggregationLogicTable(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是AggregationLogicTableVO
400	BadRequest
401	Unauthorized

状态码	描述
403	Forbidden

## 8.19.3 更新汇总表

### 功能介绍

更新汇总表。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/aggregation-logic-tables

表 8-1188 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-1189 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

表 8-1190 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	汇总表的唯一系统ID, 更新时必须填, 创建时不须填写, ID字符串。
tb_name	是	String	汇总表英文名称, 对应实际的物理表名。
tb_logic_name	是	String	汇总表的中文名, 用于展示使用。
l2_id	否	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	是	String	汇总表所属主题的ID, 必填, ID字符串。
description	否	String	汇总表描述信息。
owner	是	String	汇总表的资产责任人。
secret_type	否	String	密级类型。 枚举值: <ul style="list-style-type: none"> <li>● PUBLIC: 公开</li> <li>● SECRET: 秘密</li> <li>● CONFIDENTIAL: 机密</li> <li>● SUPER_SECRET: 绝密</li> </ul>
apply_bg	否	String	适用bg。
queue_name	否	String	dli数据连接执行sql所需的队列, 数据连接类型为DLI时必须填写。
dw_id	是	String	汇总表所在的数据连接ID, 为32位十六进制数字。
db_name	是	String	汇总表所在的数据库名。
schema	否	String	汇总表所在的Schema名, DWS类型必须填写。

参数	是否必选	参数类型	描述
dw_name	否	String	数据连接名称，只读，创建和更新时无需填写。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	否	String	表发布后，创建的数据目录技术资产guid，只读，创建和更新时无需填写。
tb_logic_guid	否	String	表发布后，创建的数据目录业务资产guid，只读，创建和更新时无需填写。
dw_type	是	String	数据连接类型，对应表所在的数据仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManagementVO object	业务版本管理，只读。
dimension_group	否	String	颗粒度ID。

参数	是否必选	参数类型	描述
time_period	否	AggregationLogicTableAttributeVO object	汇总表时间周期属性信息。
table_attributes	否	Array of AggregationLogicTableAttributeVO objects	汇总表属性信息，依据 attribute_type 判断类型。
distribute	否	String	DISTRIBUTE BY [HASH(column)  REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行 Hash，通过映射，把数据分布到指定 DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	否	String	DISTRIBUTE BY HASH column.
compression	否	String	DWS数据压缩等级，列压缩等级为no/low/middle/high，行压缩等级为no/yes。 枚举值： <ul style="list-style-type: none"> <li>"NO": 不压缩</li> <li>"YES": 压缩</li> <li>"LOW": 低等级压缩</li> <li>"MIDDLE": 中等级压缩</li> <li>"HIGH": 高等级压缩</li> </ul>
obs_location	否	String	外表路径。
pre_combine_field	否	String	版本字段。
table_type	否	String	表类型。
dlf_task_id	否	String	DLF作业ID。
quality_id	否	String	质量ID，ID字符串。
partition_conf	否	String	分区表达式。
dirty_out_switch	否	Boolean	异常数据输出开关。

参数	是否必选	参数类型	描述
dirty_out_database	否	String	异常数据输出库。
dirty_out_prefix	否	String	异常表前缀。
dirty_out_suffix	否	String	异常表后缀。
alias	否	String	别名。
configs	否	String	其他配置。
self_defined_fields	否	Array of <a href="#">SelfDefinedFieldVO</a> objects	自定义项。
api_id	否	String	API ID。
sql	否	String	汇总表绑定的SQL。
model_id	否	Long	所属模型ID。
model	否	<a href="#">WorkspaceVO</a> object	模型信息。

表 8-1191 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID, ID字符串。
tenant_id	否	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>



参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-1192 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-1193 AggregationLogicTableAttributeVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
aggregation_logic_table_id	否	String	所属汇总表ID，ID字符串。
ordinal	否	Integer	序号。
name_en	是	String	字段名。
name_ch	是	String	业务属性。

参数	是否必选	参数类型	描述
attribute_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
is_primary_key	是	Boolean	是否主键。
is_partition_key	是	Boolean	是否分区键。
not_null	否	Boolean	是否不为空。
description	否	String	描述。
data_type	是	String	字段类型。
data_type_extend	否	String	数据类型扩展字段。
ref_id	否	String	属性关联对象的id



参数	是否必选	参数类型	描述
stand_row_id	否	String	关联的数据标准的ID，ID字符串。
alias	否	String	别名。

表 8-1194 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

表 8-1195 WorkspaceVO

参数	是否必选	参数类型	描述
id	否	String	编号，ID字符串。
name	是	String	工作区名字。
description	否	String	描述。
is_physical	否	Boolean	是否为物理表。
frequent	否	Boolean	是否为常用。
top	否	Boolean	分层治理。
level	否	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	否	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_by	否	String	创建人。
update_by	否	String	更新人。

参数	是否必选	参数类型	描述
type	是	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	否	String	关联的业务分层的ID列表 { "l1Ids": [], "l2Ids": [], "l3Ids": [] }。
databases	否	Array of strings	数据库名称数组。
table_model_prefix	否	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

## 响应参数

状态码：200

表 8-1196 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：400

表 8-1197 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1198 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。

参数	参数类型	描述
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1199 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据入参，修改汇总表信息。

PUT https://{endpoint}/v2/{project\_id}/design/aggregation-logic-tables

```
{
  "tb_name": "test_create_aggregation_logic_table",
  "tb_logic_name": "test_create_aggregation_logic_table",
  "alias": "",
  "dimension_group": "0",
  "secret_type": "PUBLIC",
  "apply_bg": "",
  "l3_id": "1105525024175058944",
  "description": "无",
  "owner": "autotest_tics",
  "dw_type": "DWS",
  "dw_id": "8259af52bd294f98920ebce75a426391",
  "db_name": "bi",
  "schema": "",
  "queue_name": "default",
  "self_defined_fields": [ {
    "fd_name_ch": "aaa",
    "fd_name_en": "aaa",
    "not_null": false,
    "fd_value": ""
  } ],
  "configs": "{}",
  "table_type": "MANAGED",
  "obs_location": "",
  "table_attributes": [ {
    "id": "1227937502644600832",
    "name_ch": "col_aa",
    "name_en": "col_aa",
    "ordinal": 1,
    "is_primary_key": false,
    "is_partition_key": false,
    "not_null": false,
    "description": "",
    "data_type": "BIGINT",
    "data_type_extend": "",
    "stand_row_id": null,

```

```

"stand_row_name" : null,
"ref_id" : null,
"ref_name_ch" : null,
"quality_infos" : [ {
  "data_quality_id" : "1227937502644600833"
} ],
"secrecy_levels" : [ {
  "name" : "secrecy_level_name"
} ]
}],
"id" : "1227937502401331200",
"sql" : "",
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"partition_conf" : null
}

```

## 响应示例

**状态码：200**

Success, 返回data数据是AggregationLogicTableVO。

```

{
  "data" : {
    "value" : {
      "id" : "1227937502401331200",
      "tb_name" : "test_create_aggregation_logic_table",
      "tb_logic_name" : "test_create_aggregation_logic_table",
      "l1_id" : null,
      "l2_id" : null,
      "l3_id" : "1105525024175058944",
      "description" : "无",
      "owner" : "autotest_tics",
      "create_by" : "test_uesr",
      "queue_name" : "default",
      "dw_id" : "8259af52bd294f98920ebce75a426391",
      "db_name" : "bi",
      "tb_id" : null,
      "schema" : "",
      "dw_name" : null,
      "status" : "DRAFT",
      "tb_guid" : null,
      "tb_logic_guid" : null,
      "dw_type" : "DWS",
      "l1" : null,
      "l2" : null,
      "l3" : null,
      "create_time" : "2024-04-11T11:05:15+08:00",
      "update_time" : "2024-04-11T11:28:17.101+08:00",
      "approval_info" : null,
      "new_biz" : null,
      "dimension_group" : "0",
      "group_name" : null,
      "group_code" : null,
      "time_period" : null,
      "table_attributes" : [ {
        "id" : "1227937502644600832",
        "aggregation_logic_table_id" : null,
        "ordinal" : 1,
        "name_en" : "col_aa",
        "name_ch" : "col_aa",
        "attribute_type" : null,
        "is_primary_key" : false,
        "is_partition_key" : false,
        "not_null" : false,
        "secrecy_levels" : null,

```

```

"description": "",
"data_type": "BIGINT",
"domain_type": "NUMBER",
"data_type_extend": "",
"ref_id": null,
"ref_name_ch": null,
"ref_name_en": null,
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": [],
"alias": null
}, {
"id": "1227937502644600833",
"aggregation_logic_table_id": null,
"ordinal": 2,
"name_en": "col_bb",
"name_ch": "col_bb",
"attribute_type": null,
"is_primary_key": false,
"is_partition_key": false,
"not_null": false,
"secrecy_levels": null,
"description": "",
"data_type": "BIGINT",
"domain_type": "NUMBER",
"data_type_extend": "",
"ref_id": null,
"ref_name_ch": null,
"ref_name_en": null,
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": [],
"alias": null
}, {
"id": null,
"aggregation_logic_table_id": null,
"ordinal": 3,
"name_en": "col_cc",
"name_ch": "col_cc",
"attribute_type": null,
"is_primary_key": false,
"is_partition_key": false,
"not_null": false,
"secrecy_levels": null,
"description": "",
"data_type": "BIGINT",
"domain_type": "NUMBER",
"data_type_extend": "",
"ref_id": null,
"ref_name_ch": null,
"ref_name_en": null,
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": null,
"alias": null
}],
"physical_table": null,
"dev_physical_table": null,
"technical_asset": null,
"business_asset": null,
"meta_data_link": null,
"data_quality": null,
"dlf_task": null,
"publish_to_dlm": null,
"summary_status": null,
"distribute": null,
"distribute_column": null,
"compression": null,
"obs_location": null,

```

```
"pre_combine_field" : null,
"table_type" : "MANAGED",
"dlf_task_id" : null,
"quality_id" : null,
"reversed" : false,
"table_version" : 0,
"partition_conf" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"alias" : "",
"configs" : "{}",
"self_defined_fields" : [ {
  "fd_name_ch" : "aaa",
  "fd_name_en" : "aaa",
  "not_null" : false,
  "fd_value" : ""
}],
"api_id" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : null,
"sql" : ""
}
}
```

#### 状态码：400

##### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

#### 状态码：401

##### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

#### 状态码：403

##### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据入参，修改汇总表信息。

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateDesignAggregationLogicTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateDesignAggregationLogicTableRequest request = new
        UpdateDesignAggregationLogicTableRequest();
        AggregationLogicTableVO body = new AggregationLogicTableVO();
        List<SelfDefinedFieldVO> listbodySelfDefinedFields = new ArrayList<>();
        listbodySelfDefinedFields.add(
            new SelfDefinedFieldVO()
                .withFdNameCh("aaa")
                .withFdNameEn("aaa")
                .withNotNull(false)
                .withFdValue("")
        );
        List<QualityInfoVO> listTableAttributesQualityInfos = new ArrayList<>();
        listTableAttributesQualityInfos.add(
            new QualityInfoVO()
                .withDataQualityId("1227937502644600833")
        );
        List<SecrecyLevelVO> listTableAttributesSecrecyLevels = new ArrayList<>();
        listTableAttributesSecrecyLevels.add(
            new SecrecyLevelVO()
                .withName("secrecy_level_name")
        );
        List<AggregationLogicTableAttributeVO> listbodyTableAttributes = new ArrayList<>();
        listbodyTableAttributes.add(
            new AggregationLogicTableAttributeVO()
                .withId("1227937502644600832")
                .withOrdinal(1)
                .withNameEn("col_aa")
                .withNameCh("col_aa")
                .withIsPrimaryKey(false)
                .withIsPartitionKey(false)
                .withSecrecyLevels(listTableAttributesSecrecyLevels)
                .withNotNull(false)
                .withDescription("")
                .withDataType("BIGINT")
                .withDataTypeExtend("")
                .withQualityInfos(listTableAttributesQualityInfos)
        );
    }
}
```

```
);
body.withSql("");
body.withSelfDefinedFields(listbodySelfDefinedFields);
body.withConfigs("{}");
body.withAlias("");
body.withDirtyOutSuffix("");
body.withDirtyOutPrefix("");
body.withDirtyOutDatabase("");
body.withDirtyOutSwitch(false);
body.withTableType("MANAGED");
body.withObsLocation("");
body.withTableAttributes(listbodyTableAttributes);
body.withDimensionGroup("0");
body.withDwType("DWS");
body.withSchema("");
body.withDbName("bi");
body.withDwId("8259af52bd294f98920ebce75a426391");
body.withQueueName("default");
body.withApplyBg(AggregationLogicTableVO.ApplyBgEnum.fromValue(""));
body.withSecretType(AggregationLogicTableVO.SecretTypeEnum.fromValue("PUBLIC"));
body.withOwner("autotest_tics");
body.withDescription("无");
body.withL3Id("1105525024175058944");
body.withTbLogicName("test_create_aggregation_logic_table");
body.withTbName("test_create_aggregation_logic_table");
body.withId("1227937502401331200");
request.withBody(body);
try {
    UpdateDesignAggregationLogicTableResponse response =
client.updateDesignAggregationLogicTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据入参，修改汇总表信息。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```



```
client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateDesignAggregationLogicTableRequest()
    listSelfDefinedFieldsbody = [
        SelfDefinedFieldVO(
            fd_name_ch="aaa",
            fd_name_en="aaa",
            not_null=False,
            fd_value=""
        )
    ]
    listQualityInfosTableAttributes = [
        QualityInfoVO(
            data_quality_id="1227937502644600833"
        )
    ]
    listSecrecyLevelsTableAttributes = [
        SecrecyLevelVO(
            name="secrecy_level_name"
        )
    ]
    listTableAttributesbody = [
        AggregationLogicTableAttributeVO(
            id="1227937502644600832",
            ordinal=1,
            name_en="col_aa",
            name_ch="col_aa",
            is_primary_key=False,
            is_partition_key=False,
            secrecy_levels=listSecrecyLevelsTableAttributes,
            not_null=False,
            description="",
            data_type="BIGINT",
            data_type_extend="",
            quality_infos=listQualityInfosTableAttributes
        )
    ]
    request.body = AggregationLogicTableVO(
        sql="",
        self_defined_fields=listSelfDefinedFieldsbody,
        configs="{ }",
        alias="",
        dirty_out_suffix="",
        dirty_out_prefix="",
        dirty_out_database="",
        dirty_out_switch=False,
        table_type="MANAGED",
        obs_location="",
        table_attributes=listTableAttributesbody,
        dimension_group="0",
        dw_type="DWS",
        schema="",
        db_name="bi",
        dw_id="8259af52bd294f98920ebce75a426391",
        queue_name="default",
        apply_bg="",
        secret_type="PUBLIC",
        owner="autotest_tics",
        description="无",
        l3_id="1105525024175058944",
        tb_logic_name="test_create_aggregation_logic_table",
        tb_name="test_create_aggregation_logic_table",
        id="1227937502401331200"
    )
    response = client.update_design_aggregation_logic_table(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据入参，修改汇总表信息。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDesignAggregationLogicTableRequest{}
    fdNameChSelfDefinedFields:= "aaa"
    fdNameEnSelfDefinedFields:= "aaa"
    notNullSelfDefinedFields:= false
    fdValueSelfDefinedFields:= ""
    var listSelfDefinedFieldsbody = []model.SelfDefinedFieldVo{
        {
            FdNameCh: &fdNameChSelfDefinedFields,
            FdNameEn: &fdNameEnSelfDefinedFields,
            NotNull: &notNullSelfDefinedFields,
            FdValue: &fdValueSelfDefinedFields,
        },
    }
    var listQualityInfosTableAttributes = []model.QualityInfoVo{
        {
            DataQualityId: "1227937502644600833",
        },
    }
    var listSecrecyLevelsTableAttributes = []model.SecrecyLevelVo{
        {
            Name: "secrecy_level_name",
        },
    }
    idTableAttributes:= "1227937502644600832"
    ordinalTableAttributes:= int32(1)
    notNullTableAttributes:= false
```

```

descriptionTableAttributes:= ""
dataTypeExtendTableAttributes:= ""
var listTableAttributesbody = []model.AggregationLogicTableAttributeVo{
    {
        Id: &idTableAttributes,
        Ordinal: &ordinalTableAttributes,
        NameEn: "col_aa",
        NameCh: "col_aa",
        IsPrimaryKey: false,
        IsPartitionKey: false,
        SecrecyLevels: &listSecrecyLevelsTableAttributes,
        NotNull: &notNullTableAttributes,
        Description: &descriptionTableAttributes,
        DataType: "BIGINT",
        DataTypeExtend: &dataTypeExtendTableAttributes,
        QualityInfos: &listQualityInfosTableAttributes,
    },
}
sqlAggregationLogicTableVo:= ""
configsAggregationLogicTableVo:= "{}"
aliasAggregationLogicTableVo:= ""
dirtyOutSuffixAggregationLogicTableVo:= ""
dirtyOutPrefixAggregationLogicTableVo:= ""
dirtyOutDatabaseAggregationLogicTableVo:= ""
dirtyOutSwitchAggregationLogicTableVo:= false
tableTypeAggregationLogicTableVo:= "MANAGED"
obsLocationAggregationLogicTableVo:= ""
dimensionGroupAggregationLogicTableVo:= "0"
schemaAggregationLogicTableVo:= ""
queueNameAggregationLogicTableVo:= "default"
applyBgApplyBg:= model.GetApplyBgEnumApplyBgEnum().EMPTY
secretTypeSecretType:= model.GetSecretTypeEnumSecretTypeEnum().PUBLIC
descriptionAggregationLogicTableVo:= "无"
idAggregationLogicTableVo:= "1227937502401331200"
request.Body = &model.AggregationLogicTableVo{
    Sql: &sqlAggregationLogicTableVo,
    SelfDefinedFields: &listSelfDefinedFieldsbody,
    Configs: &configsAggregationLogicTableVo,
    Alias: &aliasAggregationLogicTableVo,
    DirtyOutSuffix: &dirtyOutSuffixAggregationLogicTableVo,
    DirtyOutPrefix: &dirtyOutPrefixAggregationLogicTableVo,
    DirtyOutDatabase: &dirtyOutDatabaseAggregationLogicTableVo,
    DirtyOutSwitch: &dirtyOutSwitchAggregationLogicTableVo,
    TableType: &tableTypeAggregationLogicTableVo,
    ObsLocation: &obsLocationAggregationLogicTableVo,
    TableAttributes: &listTableAttributesbody,
    DimensionGroup: &dimensionGroupAggregationLogicTableVo,
    DwType: "DWS",
    Schema: &schemaAggregationLogicTableVo,
    DbName: "bi",
    DwId: "8259af52bd294f98920ebce75a426391",
    QueueName: &queueNameAggregationLogicTableVo,
    ApplyBg: &applyBgApplyBg,
    SecretType: &secretTypeSecretType,
    Owner: "autotest_tics",
    Description: &descriptionAggregationLogicTableVo,
    L3Id: "1105525024175058944",
    TbLogicName: "test_create_aggregation_logic_table",
    TbName: "test_create_aggregation_logic_table",
    Id: &idAggregationLogicTableVo,
}
response, err := client.UpdateDesignAggregationLogicTable(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是AggregationLogicTableVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.19.4 删除汇总表

### 功能介绍

批量删除汇总表，只能删除状态为草稿、已线下、已驳回的表。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/aggregation-logic-tables

表 8-1200 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-1201 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1202 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-1203 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-1204 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

### 状态码：400

表 8-1205 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

### 状态码：401

表 8-1206 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

### 状态码：403

表 8-1207 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据入参，删除汇总表。

```
DELETE https://{endpoint}/v2/{project_id}/design/aggregation-logic-tables  
  
{  
  "ids": [ "1227937502401331200" ],  
  "del_types": "PHYSICAL_TABLE"  
}
```

## 响应示例

### 状态码：200

Success，返回data数据是删除个数。

```
{
  "data": {
    "value": 1
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据入参，删除汇总表。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteDesignAggregationLogicTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteDesignAggregationLogicTableRequest request = new
DeleteDesignAggregationLogicTableRequest();
IdsParam body = new IdsParam();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1227937502401331200");
body.withDelTypes(IdsParam.DelTypeEnum.fromValue("PHYSICAL_TABLE"));
body.withIds(listbodyIds);
request.withBody(body);
try {
    DeleteDesignAggregationLogicTableResponse response =
client.deleteDesignAggregationLogicTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据入参，删除汇总表。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteDesignAggregationLogicTableRequest()
        listbody = [
            "1227937502401331200"
        ]
```



```
request.body = IdsParam(  
    del_types="PHYSICAL_TABLE",  
    ids=listIdsbody  
)  
response = client.delete_design_aggregation_logic_table(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

## Go

根据入参，删除汇总表。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dataartsstudio.NewDataArtsStudioClient(  
        dataartsstudio.DataArtsStudioClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.DeleteDesignAggregationLogicTableRequest{}  
    var listIdsbody = []string{  
        "1227937502401331200",  
    }  
    delTypesIdsParam := model.GetIdsParamDelTypesEnum().PHYSICAL_TABLE  
    request.Body = &model.IdsParam{  
        DelTypes: &delTypesIdsParam,  
        Ids: listIdsbody,  
    }  
    response, err := client.DeleteDesignAggregationLogicTable(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.19.5 查看汇总表详情

### 功能介绍

通过ID查看汇总表的详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/aggregation-logic-tables/{id}

表 8-1208 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-1209 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

## 请求参数

表 8-1210 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1211 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	返回数据。

表 8-1212 data

参数	参数类型	描述
value	<a href="#">AggregationLogi cTableVO</a> object	返回的数据信息。

表 8-1213 AggregationLogicTableVO

参数	参数类型	描述
id	String	汇总表的唯一系统ID，更新时必须填，创建时不须填写，ID字符串。
tb_name	String	汇总表英文名称，对应实际的物理表名。
tb_logic_name	String	汇总表的中文名，用于展示使用。
l1_id	String	主题域分组ID，只读，创建和更新时无需填写，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	汇总表所属主题ID，必填，ID字符串。
description	String	汇总表描述信息。
owner	String	汇总表的资产责任人。
secret_type	String	密级类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLIC: 公开</li> <li>• SECRET: 秘密</li> <li>• CONFIDENTIAL: 机密</li> <li>• SUPER_SECRET: 绝密</li> </ul>
apply_bg	String	适用bg。
create_by	String	汇总表的创建人，只读，创建和更新时无需填写。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须填写。
dw_id	String	汇总表所在的数据连接ID，为32位十六进制数字。
db_name	String	汇总表所在的数据库名。
tb_id	String	汇总表创建的表ID，是服务内部ID，只读，创建和更新时无需填写
schema	String	汇总表所在的Schema名，DWS类型必须填写。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。

参数	参数类型	描述
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
tb_guid	String	表发布后，创建的数据目录技术资产 guid，只读，创建和更新时无需填写。
tb_logic_guid	String	表发布后，创建的数据目录业务资产 guid，只读，创建和更新时无需填写。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
create_time	String	创建时间，只读，创建和更新时无需填写。
update_time	String	更新时间，只读，创建和更新时无需填写。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。
dimension_group	String	颗粒度ID。
group_name	String	颗粒度名称，只读。
group_code	String	颗粒度编码，只读。

参数	参数类型	描述
time_period	<b>AggregationLogi cTableAttributeV O</b> object	汇总表时间周期属性信息。
table_attributes	Array of <b>AggregationLogi cTableAttributeV O</b> objects	汇总表属性信息，依据attribute_type判断类型。
physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
compression	String	DWS数据压缩等级，列压缩等级为no/low/middle/high，行压缩等级为no/yes。 枚举值： <ul style="list-style-type: none"> <li>• "NO": 不压缩</li> <li>• "YES": 压缩</li> <li>• "LOW": 低等级压缩</li> <li>• "MIDDLE": 中等级压缩</li> <li>• "HIGH": 高等级压缩</li> </ul>
obs_location	String	外表路径。
pre_combine_field	String	版本字段。
table_type	String	表类型。
dlf_task_id	String	DLF作业ID。
quality_id	String	质量ID，ID字符串。
reversed	Boolean	是否是逆向的，只读。

参数	参数类型	描述
table_version	Integer	为2时，表示汇总表是汇总生成的，只读。
partition_conf	String	分区表达式。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
alias	String	别名。
configs	String	其他配置。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
api_id	String	API ID。
sql	String	汇总表绑定的SQL。
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
model_id	Long	所属模型ID。
model	<b>WorkspaceVO</b> object	模型信息。

表 8-1214 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。

参数	参数类型	描述
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1215 AggregationLogicTableAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
aggregation_logic_table_id	String	所属汇总表ID，ID字符串。
ordinal	Integer	序号。
name_en	String	字段名。
name_ch	String	业务属性。

参数	参数类型	描述
attribute_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
is_primary_key	Boolean	是否主键。
is_partition_key	Boolean	是否分区键。
secrecy_levels	Array of <a href="#">SecrecyLevelVO</a> objects	密级
not_null	Boolean	是否不为空。
description	String	描述。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>• NUMBER: 数字类型</li> <li>• STRING: 字符类型</li> <li>• DATETIME: 日期类型</li> <li>• BLOB: 大对象(BLOB)</li> <li>• OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
ref_id	String	属性关联对象的id
ref_name_ch	String	属性关联对象的中文名
ref_name_en	String	属性关联对象的英文名
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。



参数	参数类型	描述
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。

表 8-1216 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1217 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1218 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1219 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-1220 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1221 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1222 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1223 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。



参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

通过ID查看汇总表的详情信息。

GET https://{endpoint}/v2/{project\_id}/design/aggregation-logic-tables/1209135365252866070

## 响应示例

**状态码：200**

Success, 返回data数据是AggregationLogicTableVO详情。

```
{
  "data": {
    "value": {
      "id": "1209135365252866070",
      "tb_name": "dws_dim_ssww",
      "tb_logic_name": "yaqiang",
      "l1_id": "1180536695448350720",
      "l2_id": null,
      "l3_id": null,
      "description": "Auto converge table.\nDerivative metrics: 1\n\nDimension Group: yaqiang/dim_ssww",
      "owner": "dgc_user",
      "create_by": "test_uesr",
      "queue_name": "default",
      "dw_id": "8259af52bd294f98920ebce75a426391",
      "db_name": "default",
      "tb_id": null,
      "schema": "",
      "dw_name": "dws",
      "status": "PUBLISHED",
      "tb_guid": "1bc46477-6cf8-42c5-8e49-cabe9e7e50c8",
      "tb_logic_guid": "c27a4a66-2bb6-4c65-b943-87594d396432",
      "dw_type": "DWS",
      "l1": "owner_test",
      "l2": null,
      "l3": null,
      "create_time": "2024-02-19T13:52:16+08:00",
      "update_time": "2024-04-19T14:21:49+08:00",
      "approval_info": {
        "id": "1230885989459886080",
        "tenant_id": null,
        "name_ch": null,
        "name_en": null,
        "biz_id": "1209135365252866070",
        "biz_type": null,
        "biz_info": null,
        "biz_info_obj": null,
        "biz_version": 0,
        "biz_status": "PUBLISHED",
        "approval_status": "APPROVED",
        "approval_type": null,
        "submit_time": "2024-04-19T14:21:29+08:00",
        "create_by": null,
        "l1": null,
        "l2": null,
        "l3": null,
        "approval_time": "2024-04-19T14:21:28+08:00",
        "approver": "test_uesr",
        "email": null,
      }
    }
  }
}
```

```

"msg": "Fast Approval OK.",
"directory_path": null
},
"new_biz": null,
"dimension_group": "0",
"group_name": null,
"group_code": null,
"time_period": null,
"table_attributes": [ {
  "id": "1209135365252866069",
  "aggregation_logic_table_id": "1209135365252866070",
  "ordinal": 1,
  "name_en": "dws_atmoic",
  "name_ch": "dws原子指标(维度DWS.hierarchies_1)",
  "attribute_type": "DERIVATIVE_INDEX",
  "is_primary_key": false,
  "is_partition_key": false,
  "not_null": false,
  "secrecy_levels": [ ],
  "description": null,
  "data_type": "STRING",
  "domain_type": "STRING",
  "data_type_extend": "",
  "ref_id": "1188156778513338368",
  "ref_name_ch": "dws原子指标(维度DWS.hierarchies_1)",
  "ref_name_en": "dws_atmoic",
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": [ ],
  "alias": null
}, {
  "id": "1209135366414688256",
  "aggregation_logic_table_id": "1209135365252866070",
  "ordinal": 2,
  "name_en": "dim_LI_dim_key",
  "name_ch": "维度DWS_维度DWS",
  "attribute_type": "SUMMARY_DIMENSION_ATTRIBUTE",
  "is_primary_key": false,
  "is_partition_key": false,
  "not_null": false,
  "secrecy_levels": [ ],
  "description": null,
  "data_type": "BIGINT",
  "domain_type": "NUMBER",
  "data_type_extend": "",
  "ref_id": "1113507850438758400#1154065454386561024.1114569975517106176",
  "ref_name_ch": "维度DWS_维度DWS",
  "ref_name_en": "dim_LI.dim_key",
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": [ ],
  "alias": null
}, {
  "id": "1209135366414688257",
  "aggregation_logic_table_id": "1209135365252866070",
  "ordinal": 3,
  "name_en": "dim_LI_li",
  "name_ch": "维度DWS_li",
  "attribute_type": "SUMMARY_DIMENSION_ATTRIBUTE",
  "is_primary_key": false,
  "is_partition_key": false,
  "not_null": true,
  "secrecy_levels": [ ],
  "description": null,
  "data_type": "BIGINT",
  "domain_type": "NUMBER",
  "data_type_extend": "",
  "ref_id": "1113507850438758400#1154065454386561024.1114569975517106177",
  "ref_name_ch": "维度DWS.li",

```

```
"ref_name_en" : "dim_LI.li",
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : [ ],
"alias" : null
}, {
  "id" : "1230885274481991680",
  "aggregation_logic_table_id" : "1209135365252866070",
  "ordinal" : 4,
  "name_en" : "order_count",
  "name_ch" : "订单数",
  "attribute_type" : "DERIVATIVE_INDEX",
  "is_primary_key" : false,
  "is_partition_key" : false,
  "not_null" : false,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : "STRING",
  "data_type_extend" : "",
  "ref_id" : "1223668870544605184",
  "ref_name_ch" : "订单数",
  "ref_name_en" : "order_count",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : [ ],
  "alias" : null
}, {
  "id" : "1230885756508184576",
  "aggregation_logic_table_id" : "1209135365252866070",
  "ordinal" : 5,
  "name_en" : "xxx7",
  "name_ch" : "xxx2",
  "attribute_type" : "DERIVED_METRIC",
  "is_primary_key" : false,
  "is_partition_key" : false,
  "not_null" : false,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : "STRING",
  "data_type_extend" : "",
  "ref_id" : "1169727308433653760",
  "ref_name_ch" : "xxx2",
  "ref_name_en" : "xxx7",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : [ ],
  "alias" : null
}, {
  "id" : "1230885835012972544",
  "aggregation_logic_table_id" : "1209135365252866070",
  "ordinal" : 6,
  "name_en" : "cdm_mysql_string_auto_col_char",
  "name_ch" : "cdm_mysql_string_auto_col_char",
  "attribute_type" : "SUMMARY_DIMENSION_ATTRIBUTE",
  "is_primary_key" : false,
  "is_partition_key" : false,
  "not_null" : false,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : "STRING",
  "data_type_extend" : "",
  "ref_id" : "raw#cdm_mysql_string_auto.col_char",
  "ref_name_ch" : "cdm_mysql_string_auto.col_char",
  "ref_name_en" : "cdm_mysql_string_auto.col_char",
  "stand_row_id" : null,
  "stand_row_name" : null,

```

```

"quality_infos" : [ ],
"alias" : null
}, {
  "id" : "1230885835012972545",
  "aggregation_logic_table_id" : "1209135365252866070",
  "ordinal" : 7,
  "name_en" : "cdm_mysql_string_auto_col_varchar",
  "name_ch" : "cdm_mysql_string_auto_col_varchar",
  "attribute_type" : "SUMMARY_DIMENSION_ATTRIBUTE",
  "is_primary_key" : false,
  "is_partition_key" : false,
  "not_null" : false,
  "secrecy_levels" : [ ],
  "description" : "",
  "data_type" : "STRING",
  "domain_type" : "STRING",
  "data_type_extend" : "",
  "ref_id" : "raw#cdm_mysql_string_auto.col_varchar",
  "ref_name_ch" : "cdm_mysql_string_auto.col_varchar",
  "ref_name_en" : "cdm_mysql_string_auto.col_varchar",
  "stand_row_id" : null,
  "stand_row_name" : null,
  "quality_infos" : [ ],
  "alias" : null
}],
"physical_table" : "NO_NEED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "CREATE_SUCCESS",
"business_asset" : "CREATE_SUCCESS",
"meta_data_link" : "UPDATE_SUCCESS",
"data_quality" : "UPDATE_SUCCESS",
"dlf_task" : "UPDATE_SUCCESS",
"publish_to_dlm" : "CREATE_FAILED",
"summary_status" : "SUMMARY_FAILED",
"distribute" : null,
"distribute_column" : null,
"compression" : null,
"obs_location" : null,
"pre_combine_field" : null,
"table_type" : "MANAGED",
"dlf_task_id" : "default_dws_dim_swww_1209135365252866070",
"quality_id" : "1230886041247125505",
"reversed" : null,
"table_version" : 0,
"partition_conf" : null,
"dirty_out_switch" : true,
"dirty_out_database" : "bi",
"dirty_out_prefix" : "dd1",
"dirty_out_suffix" : "dd2",
"alias" : null,
"configs" : "{}",
"self_defined_fields" : [ {
  "fd_name_ch" : "aaa",
  "fd_name_en" : "aaa",
  "not_null" : false,
  "fd_value" : "aaa"
}],
"api_id" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "PROD_TYPE",
"sql" : ""
}
}
}

```

状态码：400

### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowAggregationLogicTableByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAggregationLogicTableByIdRequest request = new ShowAggregationLogicTableByIdRequest();
        request.withId("{id}");
        try {
            ShowAggregationLogicTableByIdResponse response =
```

```
client.showAggregationLogicTableById(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowAggregationLogicTableByIdRequest()
        request.id = "{id}"
        response = client.show_aggregation_logic_table_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```

variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowAggregationLogicTableByIdRequest{}
request.Id = "{id}"
response, err := client.ShowAggregationLogicTableById(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是AggregationLogicTableVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.20 业务指标接口

### 8.20.1 查询业务指标信息

#### 功能介绍

通过名称、创建者、修改时间分页查找业务指标信息。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/design/biz-metrics

表 8-1224 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1225 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
owner	否	String	按负责人查询。
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
sync_status	否	String	同步状态枚举。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 同步中</li> <li>• NO_NEED: 未同步</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> </ul>



参数	是否必选	参数类型	描述
sync_key	否	Array of strings	同步任务类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• BUSINESS_ASSET: 同步业务资产</li> <li>• DATA_QUALITY: 创建质量作业</li> <li>• TECHNICAL_ASSET: 同步技术资产</li> <li>• META_DATA_LINK: 资产关联</li> <li>• PHYSICAL_TABLE: 创建表（生产环境）</li> <li>• DEV_PHYSICAL_TABLE: 创建表（开发环境）</li> <li>• DLF_TASK: 创建数据开发作业</li> <li>• MATERIALIZATION: 数值落库（码表）</li> <li>• PUBLISH_TO_DLM: 发布数据服务API</li> <li>• SUMMARY_STATUS: 整体状态</li> </ul>
biz_catalog_id	否	String	所属的业务分层的ID。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-1226 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1227 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1228 data

参数	参数类型	描述
value	<b>value</b> object	value，统一的返回结果的外层数据结构。

表 8-1229 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>BizMetricVO</b> objects	BizMetricVO信息。

表 8-1230 BizMetricVO

参数	参数类型	描述
id	String	编码，更新时必填，创建时空，ID字符串。
name	String	指标名称。
code	String	指标编码，只读。
name_alias	String	指标别名。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_catalog_id	String	归属的流程架构的ID，ID字符串。
biz_catalog_path	String	归属的流程架构路径，只读。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
data_origin	String	数据来源。
unit	String	计量单位。
time_filters	String	统计周期(时间限定)。
dimensions	String	统计维度。
general_filters	String	统计口径和修饰词。

参数	参数类型	描述
interval_type	String	刷新频率。 枚举值： <ul style="list-style-type: none"> <li>• MINUTE: 每分钟</li> <li>• HOUR: 每小时</li> <li>• DAY: 每天</li> <li>• WEEK: 每周</li> <li>• MONTH: 每月</li> <li>• YEAR: 每年</li> <li>• REAL_TIME: 实时</li> <li>• HALF_HOUR: 每半小时</li> <li>• QUART: 每15分钟</li> <li>• DOUBLE_WEEK: 每两周</li> <li>• HALF_YEAR: 每半年</li> <li>• HALF_DAY: 每半天</li> </ul>
apply_scenario	String	应用场景。
technical_metric	String	关联技术指标，ID字符串。
technical_metric_name	String	关联技术指标名称，只读。

参数	参数类型	描述
technical_metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
measure	String	度量对象。
owner	String	指标责任人。
owner_department	String	指标管理部门。
destination	String	设置目的。
guid	String	资产同步后的guid，只读。
definition	String	指标定义。
expression	String	计算公式。
remark	String	备注。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。



参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
biz_metric	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项

表 8-1231 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。

参数	参数类型	描述
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1232 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1233 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-1234 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1235 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1236 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。



## 请求示例

查询前十条业务指标数据，返回业务指标集合。

```
GET https://{endpoint}/v2/{project_id}/design/biz-metrics?offset=1&limit=10
```

## 响应示例

状态码：200

Success，返回data数据是BizMetricVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1193199235202428928",
        "name": "test_czh_001",
        "code": "ZB000000",
        "name_alias": null,
        "biz_type": null,
        "status": "PUBLISHED",
        "biz_catalog_id": "1193122318579281920",
        "biz_catalog_path": "test_czh_001",
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "data_origin": null,
        "unit": null,
        "time_filters": "天",
        "dimensions": "",
        "general_filters": null,
        "interval_type": "MONTH",
        "apply_scenario": null,
        "technical_metric": null,
        "technical_metric_name": null,
        "technical_metric_type": null,
        "measure": null,
        "owner": "test_uesr",
        "owner_department": "sss",
        "destination": "测试",
        "guid": "ead2dc91-e81d-4590-94e3-5f7f69dd85b5",
        "definition": "没有特殊含义",
        "expression": "海伦凯勒公式",
        "remark": null,
        "approval_info": {
          "id": "1193199265908928512",
          "tenant_id": null,
          "name_ch": null,
          "name_en": null,
          "biz_id": "1193199235202428928",
          "biz_type": null,
          "biz_info": null,
          "biz_info_obj": null,
          "biz_version": 0,
          "biz_status": "PUBLISHED",
          "approval_status": "APPROVED",
          "approval_type": null,
          "submit_time": "2024-01-06T14:27:54+08:00",
          "create_by": null,
          "l1": null,
          "l2": null,
          "l3": null,
          "approval_time": "2024-01-15T20:27:37+08:00",
          "approver": "test_uesr",
          "email": null,
          "msg": "1",
          "directory_path": null
        }
      }
    ]
  }
}
```

```

    },
    "new_biz" : null,
    "create_time" : "2024-01-06T14:27:46+08:00",
    "update_time" : "2024-01-15T20:27:37+08:00",
    "l1" : "test_czh_001",
    "l2" : null,
    "l3" : null,
    "biz_metric" : "CREATE_SUCCESS",
    "summary_status" : "SUMMARY_SUCCESS",
    "self_defined_fields" : null
  }
}
}
}
}

```

#### 状态码：400

##### BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

#### 状态码：401

##### Unauthorized

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}

```

#### 状态码：403

##### Forbidden

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListBizMetricsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    }
}

```

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListBizMetricsRequest request = new ListBizMetricsRequest();
try {
    ListBizMetricsResponse response = client.listBizMetrics(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBizMetricsRequest()
        response = client.list_biz_metrics(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBizMetricsRequest{}
    response, err := client.ListBizMetrics(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是BizMetricVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.20.2 创建业务指标

### 功能介绍

创建业务指标。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/biz-metrics

表 8-1237 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-1238 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1239 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码，更新时必填，创建时空，ID字符串。
name	是	String	指标名称。
name_alias	否	String	指标别名。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_catalog_id	是	String	归属的流程架构的ID，ID字符串。
data_origin	否	String	数据来源。
unit	否	String	计量单位。
time_filters	是	String	统计周期(时间限定)。
dimensions	否	String	统计维度。
general_filters	否	String	统计口径和修饰词。



参数	是否必选	参数类型	描述
interval_type	是	String	刷新频率。 枚举值： <ul style="list-style-type: none"> <li>• MINUTE: 每分钟</li> <li>• HOUR: 每小时</li> <li>• DAY: 每天</li> <li>• WEEK: 每周</li> <li>• MONTH: 每月</li> <li>• YEAR: 每年</li> <li>• REAL_TIME: 实时</li> <li>• HALF_HOUR: 每半小时</li> <li>• QUART: 每15分钟</li> <li>• DOUBLE_WEEK: 每两周</li> <li>• HALF_YEAR: 每半年</li> <li>• HALF_DAY: 每半天</li> </ul>
apply_scenario	否	String	应用场景。
technical_metric	否	String	关联技术指标, ID字符串。

参数	是否必选	参数类型	描述
technical_metric_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
measure	否	String	度量对象。
owner	是	String	指标责任人。
owner_department	是	String	指标管理部门。
destination	是	String	设置目的。
definition	是	String	指标定义。
expression	是	String	计算公式。
remark	否	String	备注。

参数	是否必选	参数类型	描述
approval_info	否	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	<b>BizVersionMa nageVO</b> object	业务版本管理,只读。
l1	否	String	主题域分组中文名,只读,创建和更新时无需填写。
l2	否	String	主题域中文名,只读,创建和更新时无需填写。
l3	否	String	业务对象中文名,只读,创建和更新时无需填写。
self_defined_fi elds	否	Array of <b>SelfDefinedFi eldVO</b> objects	自定义项

表 8-1240 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID, ID字符串。
tenant_id	否	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-1241 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。



参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-1242 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

## 响应参数

状态码：200

表 8-1243 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1244 data

参数	参数类型	描述
value	<b>BizMetricVO</b> object	value，统一的返回结果的外层数据结构。

表 8-1245 BizMetricVO

参数	参数类型	描述
id	String	编码，更新时必填，创建时空，ID字符串。
name	String	指标名称。
code	String	指标编码，只读。
name_alias	String	指标别名。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_catalog_id	String	归属的流程架构的ID，ID字符串。
biz_catalog_path	String	归属的流程架构路径，只读。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
data_origin	String	数据来源。
unit	String	计量单位。
time_filters	String	统计周期(时间限定)。
dimensions	String	统计维度。
general_filters	String	统计口径和修饰词。

参数	参数类型	描述
interval_type	String	刷新频率。 枚举值： <ul style="list-style-type: none"> <li>• MINUTE: 每分钟</li> <li>• HOUR: 每小时</li> <li>• DAY: 每天</li> <li>• WEEK: 每周</li> <li>• MONTH: 每月</li> <li>• YEAR: 每年</li> <li>• REAL_TIME: 实时</li> <li>• HALF_HOUR: 每半小时</li> <li>• QUART: 每15分钟</li> <li>• DOUBLE_WEEK: 每两周</li> <li>• HALF_YEAR: 每半年</li> <li>• HALF_DAY: 每半天</li> </ul>
apply_scenario	String	应用场景。
technical_metric	String	关联技术指标，ID字符串。
technical_metric_name	String	关联技术指标名称，只读。

参数	参数类型	描述
technical_metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
measure	String	度量对象。
owner	String	指标责任人。
owner_department	String	指标管理部门。
destination	String	设置目的。
guid	String	资产同步后的guid，只读。
definition	String	指标定义。
expression	String	计算公式。
remark	String	备注。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
biz_metric	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项

表 8-1246 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。

参数	参数类型	描述
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1247 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-1248 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-1249 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1250 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1251 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数进行业务指标的创建。

```
POST https://{endpoint}/v2/{project_id}/design/biz-metrics

{
  "code": null,
  "biz_catalog_id": "1200479561188536320",
  "name": "test_create_biz_metrics",
  "name_alias": "",
  "destination": "业务人员对此业务指标的设置目的进行描述",
  "definition": "比如：按月对数据库某张表的某个字段进行均值计算。",
  "expression": "avg(table.column)",
  "data_origin": "",
  "unit": "",
  "time_filters": "月度",
  "dimensions": "test_czh_0305_001",
  "general_filters": "",
  "interval_type": "HOUR",
  "apply_scenario": "",
  "technical_metric": "",
  "measure": "",
  "owner_department": "data owner dept",
  "owner": "test_uesr",
  "remark": "该业务指标有啥需要注意的地方。",
  "self_defined_fields": []
}
```

## 响应示例

**状态码：200**

Success, 返回data数据是BizMetricVO。

```
{
  "data": {
    "value": {
      "id": "1231249989901135872",
      "name": "test_create_biz_metrics",
      "code": "ZB000001",
      "name_alias": "",
      "biz_type": null,
      "status": "DRAFT",
      "biz_catalog_id": "1200479561188536320",
      "biz_catalog_path": null,
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "data_origin": "",
      "unit": "",
      "time_filters": "月度",
      "dimensions": "test_czh_0305_001",
      "general_filters": "",
      "interval_type": "HOUR",
      "apply_scenario": "",
      "technical_metric": null,
      "technical_metric_name": null,
      "technical_metric_type": null,
      "measure": "",
      "owner": "test_uesr",
      "owner_department": "data owner dept",
      "destination": "业务人员对此业务指标的设置目的进行描述",
      "guid": null,
      "definition": "比如：按月对数据库某张表的某个字段进行均值计算。",
      "expression": "avg(table.column)",
      "remark": "该业务指标有啥需要注意的地方。",
      "approval_info": null,
      "new_biz": null,
    }
  }
}
```

```
"create_time" : "2024-04-20T14:27:53.111+08:00",
"update_time" : "2024-04-20T14:27:53.111+08:00",
"l1" : null,
"l2" : null,
"l3" : null,
"biz_metric" : null,
"summary_status" : null,
"self_defined_fields" : [ ]
}
}
```

### 状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数进行业务指标的创建。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateBizMetricSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
CreateBizMetricRequest request = new CreateBizMetricRequest();
BizMetricVO body = new BizMetricVO();
body.withRemark("该业务指标有啥需要注意的地方。");
body.withExpression("avg(table.column)");
body.withDefinition("比如:按月对数据库某张表的某个字段进行均值计算。");
body.withDestination("业务人员对此业务指标的设置目的进行描述");
body.withOwnerDepartment("data owner dept");
body.withOwner("test_uesr");
body.withMeasure("");
body.withTechnicalMetric("");
body.withApplyScenario("");
body.withIntervalType(BizMetricVO.IntervalTypeEnum.fromValue("HOUR"));
body.withGeneralFilters("");
body.withDimensions("test_czh_0305_001");
body.withTimeFilters("月度");
body.withUnit("");
body.withDataOrigin("");
body.withBizCatalogId("1200479561188536320");
body.withNameAlias("");
body.withName("test_create_biz_metrics");
request.withBody(body);
try {
    CreateBizMetricResponse response = client.createBizMetric(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据请求参数进行业务指标的创建。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateBizMetricRequest()
    request.body = BizMetricVO(
        remark="该业务指标有啥需要注意的地方。",
        expression="avg(table.column)",
        definition="比如:按月对数据库某张表的某个字段进行均值计算。",
        destination="业务人员对此业务指标的设置目的进行描述",
        owner_department="data owner dept",
        owner="test_uesr",
        measure="",
        technical_metric="",
        apply_scenario="",
        interval_type="HOUR",
        general_filters="",
        dimensions="test_czh_0305_001",
        time_filters="月度",
        unit="",
        data_origin="",
        biz_catalog_id="1200479561188536320",
        name_alias="",
        name="test_create_biz_metrics"
    )
    response = client.create_biz_metric(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据请求参数进行业务指标的创建。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```

```

WithSk(sk).
WithProjectId(projectId).
Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateBizMetricRequest{
    remarkBizMetricVo:= "该业务指标有啥需要注意的地方。"
    measureBizMetricVo:= ""
    technicalMetricBizMetricVo:= ""
    applyScenarioBizMetricVo:= ""
    generalFiltersBizMetricVo:= ""
    dimensionsBizMetricVo:= "test_czh_0305_001"
    unitBizMetricVo:= ""
    dataOriginBizMetricVo:= ""
    nameAliasBizMetricVo:= ""
    request.Body = &model.BizMetricVo{
        Remark: &remarkBizMetricVo,
        Expression: "avg(table.column)",
        Definition: "比如:按月对数据库某张表的某个字段进行均值计算。",
        Destination: "业务人员对此业务指标的设置目的进行描述",
        OwnerDepartment: "data owner dept",
        Owner: "test_uesr",
        Measure: &measureBizMetricVo,
        TechnicalMetric: &technicalMetricBizMetricVo,
        ApplyScenario: &applyScenarioBizMetricVo,
        IntervalType: model.GetBizMetricVoIntervalTypeEnum().HOURL,
        GeneralFilters: &generalFiltersBizMetricVo,
        Dimensions: &dimensionsBizMetricVo,
        TimeFilters: "月度",
        Unit: &unitBizMetricVo,
        DataOrigin: &dataOriginBizMetricVo,
        BizCatalogId: "1200479561188536320",
        NameAlias: &nameAliasBizMetricVo,
        Name: "test_create_biz_metrics",
    }
}
response, err := client.CreateBizMetric(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是BizMetricVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.20.3 更新业务指标

### 功能介绍

更新业务指标。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/biz-metrics

表 8-1252 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-1253 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1254 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码，更新时必填，创建时空，ID字符串。
name	是	String	指标名称。
name_alias	否	String	指标别名。



参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_catalog_id	是	String	归属的流程架构的ID，ID字符串。
data_origin	否	String	数据来源。
unit	否	String	计量单位。
time_filters	是	String	统计周期(时间限定)。
dimensions	否	String	统计维度。
general_filters	否	String	统计口径和修饰词。

参数	是否必选	参数类型	描述
interval_type	是	String	刷新频率。 枚举值： <ul style="list-style-type: none"> <li>• MINUTE: 每分钟</li> <li>• HOUR: 每小时</li> <li>• DAY: 每天</li> <li>• WEEK: 每周</li> <li>• MONTH: 每月</li> <li>• YEAR: 每年</li> <li>• REAL_TIME: 实时</li> <li>• HALF_HOUR: 每半小时</li> <li>• QUART: 每15分钟</li> <li>• DOUBLE_WEEK: 每两周</li> <li>• HALF_YEAR: 每半年</li> <li>• HALF_DAY: 每半天</li> </ul>
apply_scenario	否	String	应用场景。
technical_metric	否	String	关联技术指标, ID字符串。

参数	是否必选	参数类型	描述
technical_metric_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
measure	否	String	度量对象。
owner	是	String	指标责任人。
owner_department	是	String	指标管理部门。
destination	是	String	设置目的。
definition	是	String	指标定义。
expression	是	String	计算公式。
remark	否	String	备注。

参数	是否必选	参数类型	描述
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManageVO object	业务版本管理,只读。
l1	否	String	主题域分组中文名,只读,创建和更新时无需填写。
l2	否	String	主题域中文名,只读,创建和更新时无需填写。
l3	否	String	业务对象中文名,只读,创建和更新时无需填写。
self_defined_fields	否	Array of SelfDefinedFieldVO objects	自定义项

表 8-1255 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID, ID字符串。
tenant_id	否	String	项目ID, 获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。



参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-1256 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>● AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>● ATOMIC_INDEX: 原子指标</li> <li>● ATOMIC_METRIC: 原子指标 (新)</li> <li>● BIZ_CATALOG: 流程架构目录</li> <li>● BIZ_METRIC: 业务指标</li> <li>● CODE_TABLE: 码表</li> <li>● COMMON_CONDITION: 通用限定</li> <li>● COMPOSITE_METRIC: 复合指标 (新)</li> <li>● COMPOUND_METRIC: 复合指标</li> <li>● CONDITION_GROUP: 限定分组</li> <li>● DEGENERATE_DIMENSION: 退化维度</li> <li>● DERIVATIVE_INDEX: 衍生指标</li> <li>● DERIVED_METRIC: 衍生指标 (新)</li> <li>● DIMENSION: 维度</li> <li>● DIMENSION_ATTRIBUTE: 维度属性</li> <li>● DIMENSION_HIERARCHIES: 维度层级</li> <li>● DIMENSION_LOGIC_TABLE: 维度表</li> <li>● DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>● DIRECTORY: 目录</li> <li>● FACT_ATTRIBUTE: 事实表属性</li> <li>● FACT_DIMENSION: 事实表维度</li> <li>● FACT_LOGIC_TABLE: 事实表</li> <li>● FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-1257 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

## 响应参数

状态码：200

表 8-1258 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1259 data

参数	参数类型	描述
value	Object	value，统一的返回结果的外层数据结构。

表 8-1260 BizMetricVO

参数	参数类型	描述
id	String	编码，更新时必填，创建时空，ID字符串。
name	String	指标名称。
code	String	指标编码，只读。
name_alias	String	指标别名。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_catalog_id	String	归属的流程架构的ID，ID字符串。
biz_catalog_path	String	归属的流程架构路径，只读。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
data_origin	String	数据来源。
unit	String	计量单位。
time_filters	String	统计周期(时间限定)。
dimensions	String	统计维度。
general_filters	String	统计口径和修饰词。



参数	参数类型	描述
interval_type	String	刷新频率。 枚举值： <ul style="list-style-type: none"> <li>• MINUTE: 每分钟</li> <li>• HOUR: 每小时</li> <li>• DAY: 每天</li> <li>• WEEK: 每周</li> <li>• MONTH: 每月</li> <li>• YEAR: 每年</li> <li>• REAL_TIME: 实时</li> <li>• HALF_HOUR: 每半小时</li> <li>• QUART: 每15分钟</li> <li>• DOUBLE_WEEK: 每两周</li> <li>• HALF_YEAR: 每半年</li> <li>• HALF_DAY: 每半天</li> </ul>
apply_scenario	String	应用场景。
technical_metric	String	关联技术指标，ID字符串。
technical_metric_name	String	关联技术指标名称，只读。

参数	参数类型	描述
technical_metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
measure	String	度量对象。
owner	String	指标责任人。
owner_department	String	指标管理部门。
destination	String	设置目的。
guid	String	资产同步后的guid，只读。
definition	String	指标定义。
expression	String	计算公式。
remark	String	备注。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
biz_metric	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项

表 8-1261 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。

参数	参数类型	描述
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1262 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1263 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-1264 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1265 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1266 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数进行业务指标的修改。

```
PUT https://{endpoint}/v2/{project_id}/design/biz-metrics
{
  "code": "ZB000001",
  "biz_catalog_id": "1200479561188536320",
  "name": "test_update_biz_metrics",
  "name_alias": "",
  "destination": "业务人员对此业务指标的设置目的进行描述",
  "definition": "比如：按月对数据库某张表的某个字段进行均值计算。",
  "expression": "avg(table.column)",
  "data_origin": "",
  "unit": "",
  "time_filters": "月度",
  "dimensions": "test_czh_0305_001",
  "general_filters": "",
  "interval_type": "HOUR",
  "apply_scenario": "",
  "technical_metric": "",
  "measure": "",
  "owner_department": "data owner dept",
  "owner": "test_uesr",
  "remark": "该业务指标有啥需要注意的地方。",
  "self_defined_fields": [],
  "id": "1231249989901135872"
}
```

## 响应示例

**状态码：200**

Success, 返回data数据是BizMetricVO。

```
{
  "data": {
    "value": {
      "id": "1231249989901135872",
      "name": "test_update_biz_metrics",
      "code": "ZB000001",
      "name_alias": "",
      "biz_type": null,
      "status": "DRAFT",
      "biz_catalog_id": "1200479561188536320",
      "biz_catalog_path": null,
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "data_origin": "",
      "unit": "",
      "time_filters": "月度",
      "dimensions": "test_czh_0305_001",
      "general_filters": "",
      "interval_type": "HOUR",
      "apply_scenario": "",
      "technical_metric": null,
      "technical_metric_name": null,
      "technical_metric_type": null,
      "measure": "",
      "owner": "test_uesr",
      "owner_department": "data owner dept",
      "destination": "业务人员对此业务指标的设置目的进行描述",
      "guid": null,
      "definition": "比如：按月对数据库某张表的某个字段进行均值计算。",
      "expression": "avg(table.column)",
      "remark": "该业务指标有啥需要注意的地方。",
      "approval_info": null,
    }
  }
}
```

```
"new_biz" : null,  
"create_time" : "2024-04-20T14:27:53+08:00",  
"update_time" : "2024-04-20T14:31:44.232+08:00",  
"l1" : null,  
"l2" : null,  
"l3" : null,  
"biz_metric" : null,  
"summary_status" : null,  
"self_defined_fields" : [ ]  
}  
}  
}
```

**状态码: 400**

BadRequest

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user request is illegal."  
}
```

**状态码: 401**

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

**状态码: 403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数进行业务指标的修改。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class UpdateBizMetricSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateBizMetricRequest request = new UpdateBizMetricRequest();
BizMetricVO body = new BizMetricVO();
body.withRemark("该业务指标有啥需要注意的地方。");
body.withExpression("avg(table.column)");
body.withDefinition("比如:按月对数据库某张表的某个字段进行均值计算。");
body.withDestination("业务人员对此业务指标的设置目的进行描述");
body.withOwnerDepartment("data owner dept");
body.withOwner("test_uesr");
body.withMeasure("");
body.withTechnicalMetric("");
body.withApplyScenario("");
body.withIntervalType(BizMetricVO.IntervalTypeEnum.fromValue("HOUR"));
body.withGeneralFilters("");
body.withDimensions("test_czh_0305_001");
body.withTimeFilters("月度");
body.withUnit("");
body.withDataOrigin("");
body.withBizCatalogId("1200479561188536320");
body.withNameAlias("");
body.withCode("ZB000001");
body.withName("test_update_biz_metrics");
body.withId("1231249989901135872");
request.withBody(body);
try {
    UpdateBizMetricResponse response = client.updateBizMetric(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据请求参数进行业务指标的修改。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateBizMetricRequest()
    request.body = BizMetricVO(
        remark="该业务指标有啥需要注意的地方。",
        expression="avg(table.column)",
        definition="比如:按月对数据库某张表的某个字段进行均值计算。",
        destination="业务人员对此业务指标的设置目的进行描述",
        owner_department="data owner dept",
        owner="test_uesr",
        measure="",
        technical_metric="",
        apply_scenario="",
        interval_type="HOUR",
        general_filters="",
        dimensions="test_czh_0305_001",
        time_filters="月度",
        unit="",
        data_origin="",
        biz_catalog_id="1200479561188536320",
        name_alias="",
        code="ZB000001",
        name="test_update_biz_metrics",
        id="1231249989901135872"
    )
    response = client.update_biz_metric(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据请求参数进行业务指标的修改。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateBizMetricRequest{
    remarkBizMetricVo:= "该业务指标有啥需要注意的地方。"
    measureBizMetricVo:= ""
    technicalMetricBizMetricVo:= ""
    applyScenarioBizMetricVo:= ""
    generalFiltersBizMetricVo:= ""
    dimensionsBizMetricVo:= "test_czh_0305_001"
    unitBizMetricVo:= ""
    dataOriginBizMetricVo:= ""
    nameAliasBizMetricVo:= ""
    codeBizMetricVo:= "ZB000001"
    idBizMetricVo:= "1231249989901135872"
    request.Body = &model.BizMetricVo{
        Remark: &remarkBizMetricVo,
        Expression: "avg(table.column)",
        Definition: "比如:按月对数据库某张表的某个字段进行均值计算。",
        Destination: "业务人员对此业务指标的设置目的进行描述",
        OwnerDepartment: "data owner dept",
        Owner: "test_uesr",
        Measure: &measureBizMetricVo,
        TechnicalMetric: &technicalMetricBizMetricVo,
        ApplyScenario: &applyScenarioBizMetricVo,
        IntervalType: model.GetBizMetricVoIntervalTypeEnum().HOURL,
        GeneralFilters: &generalFiltersBizMetricVo,
        Dimensions: &dimensionsBizMetricVo,
        TimeFilters: "月度",
        Unit: &unitBizMetricVo,
        DataOrigin: &dataOriginBizMetricVo,
        BizCatalogId: "1200479561188536320",
        NameAlias: &nameAliasBizMetricVo,
        Code: &codeBizMetricVo,
        Name: "test_update_biz_metrics",
        Id: &idBizMetricVo,
    }
}
response, err := client.UpdateBizMetric(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。



## 状态码

状态码	描述
200	Success, 返回data数据是BizMetricVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.20.4 删除业务指标

### 功能介绍

删除业务指标。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/biz-metrics

表 8-1267 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-1268 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 ( 响应消息头中X-Subject-Token的值 ) 使用Token认证时必选。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1269 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200

表 8-1270 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-1271 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

状态码：400

表 8-1272 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1273 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1274 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据行业务指标的ID删除业务指标。

```
DELETE https://{endpoint}/v2/{project_id}/design/biz-metrics
{
  "ids" : [ "1231252097002024960" ]
}
```

## 响应示例

状态码：200

Success，返回data数据是删除个数。

```
{
  "value" : 1
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据行业务指标的ID删除业务指标。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteBizMetricSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
DeleteBizMetricRequest request = new DeleteBizMetricRequest();
IdsParam body = new IdsParam();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1231252097002024960");
body.withIds(listbodyIds);
request.withBody(body);
try {
    DeleteBizMetricResponse response = client.deleteBizMetric(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据行业务指标的ID删除业务指标。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteBizMetricRequest()
        listbody = [
            "1231252097002024960"
        ]
        request.body = IdsParam(
            ids=listbody
        )
        response = client.delete_biz_metric(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

根据行业务指标的ID删除业务指标。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteBizMetricRequest{}
    var listIdsbody = []string{
        "1231252097002024960",
    }
    request.Body = &model.IdsParam{
        Ids: listIdsbody,
    }
    response, err := client.DeleteBizMetric(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数。

状态码	描述
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.20.5 查看指标详情

### 功能介绍

通过ID查看指标的详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/biz-metrics/{id}

表 8-1275 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-1276 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

## 请求参数

表 8-1277 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1278 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-1279 data

参数	参数类型	描述
value	<a href="#">BizMetricVO</a> object	value，统一的返回结果的外层数据结构。



表 8-1280 BizMetricVO

参数	参数类型	描述
id	String	编码，更新时必填，创建时空，ID字符串。
name	String	指标名称。
code	String	指标编码，只读。
name_alias	String	指标别名。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_catalog_id	String	归属的流程架构的ID，ID字符串。
biz_catalog_path	String	归属的流程架构路径，只读。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
data_origin	String	数据来源。
unit	String	计量单位。
time_filters	String	统计周期(时间限定)。
dimensions	String	统计维度。
general_filters	String	统计口径和修饰词。

参数	参数类型	描述
interval_type	String	刷新频率。 枚举值： <ul style="list-style-type: none"> <li>• MINUTE: 每分钟</li> <li>• HOUR: 每小时</li> <li>• DAY: 每天</li> <li>• WEEK: 每周</li> <li>• MONTH: 每月</li> <li>• YEAR: 每年</li> <li>• REAL_TIME: 实时</li> <li>• HALF_HOUR: 每半小时</li> <li>• QUART: 每15分钟</li> <li>• DOUBLE_WEEK: 每两周</li> <li>• HALF_YEAR: 每半年</li> <li>• HALF_DAY: 每半天</li> </ul>
apply_scenario	String	应用场景。
technical_metric	String	关联技术指标，ID字符串。
technical_metric_name	String	关联技术指标名称，只读。

参数	参数类型	描述
technical_metric_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
measure	String	度量对象。
owner	String	指标责任人。
owner_department	String	指标管理部门。
destination	String	设置目的。
guid	String	资产同步后的guid，只读。
definition	String	指标定义。
expression	String	计算公式。
remark	String	备注。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManagementVO</b> object	业务版本管理，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。

参数	参数类型	描述
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
biz_metric	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项

表 8-1281 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。

参数	参数类型	描述
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID, ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1282 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1283 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-1284 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1285 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1286 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据业务指标ID，查询业务指标详情。

```
GET https://{endpoint}/v2/{project_id}/design/biz-metrics/1231249989901135872
```

## 响应示例

**状态码：200**

Success，返回data数据是BizMetricVO详情。

```
{
  "data": {
    "value": {
      "id": "1231249989901135872",
      "name": "test_update_biz_metrics",
      "code": "ZB000001",
      "name_alias": "",
      "biz_type": null,
      "status": "DRAFT",
      "biz_catalog_id": "1200479561188536320",
      "biz_catalog_path": "aaa001",
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "data_origin": "",
      "unit": "",
      "time_filters": "月度",
      "dimensions": "test_czh_0305_001",
      "general_filters": "",
      "interval_type": "HOUR",
      "apply_scenario": "",
      "technical_metric": null,
      "technical_metric_name": null,
      "technical_metric_type": null,
      "measure": "",
      "owner": "test_uesr",
      "owner_department": "data owner dept",
      "destination": "业务人员对此业务指标的设置目的进行描述",
      "guid": null,
      "definition": "比如：按月对数据库某张表的某个字段进行均值计算。",
      "expression": "avg(table.column)",
      "remark": "该业务指标有啥需要注意的地方。",
      "approval_info": null,
      "new_biz": null,
      "create_time": "2024-04-20T14:27:53+08:00",
      "update_time": "2024-04-20T14:31:44+08:00",
      "l1": "aaa001",
      "l2": null,
      "l3": null,
      "biz_metric": null,
      "summary_status": null,
      "self_defined_fields": [ ]
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

## Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

## 状态码：403

## Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowBizMetricByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))
            .build();
        ShowBizMetricByIdRequest request = new ShowBizMetricByIdRequest();
        request.withId("{id}");
        try {
            ShowBizMetricByIdResponse response = client.showBizMetricById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
        }
    }
}
```



```
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowBizMetricByIdRequest()
        request.id = "{id}"
        response = client.show_biz_metric_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```

```
WithProjectId(projectId).
Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowBizMetricByIdRequest{}
request.Id = "{id}"
response, err := client.ShowBizMetricById(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是BizMetricVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.20.6 查看指标维度信息

### 功能介绍

查看指标维度信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/biz-metrics/dimensions

表 8-1287 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1288 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-1289 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1290 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1291 data

参数	参数类型	描述
value	Array of strings	value，统一的返回结果的外层数据结构。

**状态码：400**

表 8-1292 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1293 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1294 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

查看指标维度信息。

```
GET https://{endpoint}/v2/{project_id}/design/biz-metrics/dimensions
```

## 响应示例

**状态码：200**

Success, 返回data数据是维度名称数组。

```
{
  "data": {
    "value": [ "费率代码", "日期维度", "付款方式", "供应商" ]
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListBizMetricDimensionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBizMetricDimensionsRequest request = new ListBizMetricDimensionsRequest();
        try {
            ListBizMetricDimensionsResponse response = client.listBizMetricDimensions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
```

```

        .build()

    try:
        request = ListBizMetricDimensionsRequest()
        response = client.list_biz_metric_dimensions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBizMetricDimensionsRequest{}
    response, err := client.ListBizMetricDimensions(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是维度名称数组。

状态码	描述
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.20.7 查看指标责任人信息

### 功能介绍

查看指标责任人信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/biz-metrics/owners

表 8-1295 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1296 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。



## 请求参数

表 8-1297 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1298 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：400

表 8-1299 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

**表 8-1300 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-1301 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查看业务指标责任人集合。

GET https://{endpoint}/v2/{project\_id}/design/biz-metrics/owners

## 响应示例

**状态码：200**

Success，返回data数据是指标责任人名称数组。

```
{
  "data": {
    "value": [ "test_uesr" ]
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

## Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListBizMetricOwnersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBizMetricOwnersRequest request = new ListBizMetricOwnersRequest();
        try {
            ListBizMetricOwnersResponse response = client.listBizMetricOwners(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBizMetricOwnersRequest()
        response = client.list_biz_metric_owners(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBizMetricOwnersRequest{}
    response, err := client.ListBizMetricOwners(request)
```

```

if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是指标责任人名称数组。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.20.8 获取指标关联信息

### 功能介绍

获取当前指标图谱。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/metric-relations/{id}

表 8-1302 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-1303 Query 参数

参数	是否必选	参数类型	描述
biz_type	是	String	指标类型。

## 请求参数

表 8-1304 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1305 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	data，统一的返回结果的最外层数据结构。

表 8-1306 data

参数	参数类型	描述
value	value object	value, 统一的返回结果的外层数据结构。

表 8-1307 value

参数	参数类型	描述
all	Array of objects	所有的业务指标信息。
links	Object	指标关联。
groups	Object	分组。
total	Integer	总数。

状态码: 400

表 8-1308 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码: 401

表 8-1309 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DS.6000, 请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码: 403

表 8-1310 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据指标ID进行指标图谱查询。

GET https://{endpoint}/v2/{project\_id}/design/metric-relations/1193199235202428928?biz\_type=BIZ\_METRIC

## 响应示例

**状态码：200**

Success，返回data数据是MetricRelationVOList。

```
{
  "data": {
    "value": {
      "all": [ {
        "vo": {
          "id": "1014197025488699392",
          "name_en": "zonghe8830",
          "name_ch": "zonghe8830(testweidu830)",
          "description": "",
          "create_by": "abc",
          "data_type": "STRING",
          "l1_id": null,
          "l2_id": null,
          "l3_id": "767858832738234368",
          "status": "DRAFT",
          "atomic_index_id": "1014195214723756032",
          "time_condition_id": null,
          "time_field_id": null,
          "time_field_name": null,
          "common_conditions": [ ],
          "dimension_groups": [ {
            "group_id": "1014194851576721408",
            "role": null,
            "dimension_id": "1014194851576721408",
            "hierarchies_id": null,
            "ordinal": 1,
            "group_name": "testweidu830",
            "group_code": "testweidu830",
            "biz_type": "DIMENSION",
            "hierarchies": null,
            "l1": null,
            "l2": null,
            "l3": null,
            "l1_id": null,
            "l2_id": null,
            "l3_id": null,
            "dw_type": null,
            "id": "1014197025585168384"
          } ],
          "monitor": null,
          "atomic_index": {
            "id": "1014195214723756032",
```



```

"name_en" : "zonghe8830",
"name_ch" : "zonghe8830",
"description" : "",
"create_by" : null,
"cal_exp" : "sum(${1014195020200325120}) ",
"cal_fn_ids" : null,
"l1_id" : null,
"l2_id" : null,
"l3_id" : null,
"table_id" : "1014195020183547904",
"tb_name" : null,
"dw_type" : null,
"field_ids" : [ "1014195020200325120" ],
"field_names" : null,
"status" : "DRAFT",
"biz_type" : "FACT_LOGIC_TABLE",
"create_time" : "2022-09-02T09:56:31.536+08:00",
"update_time" : "2022-09-02T09:56:31.536+08:00",
"l1" : null,
"l2" : null,
"l3" : null,
"approval_info" : null,
"new_biz" : null
},
"time_condition_name" : null,
"create_time" : "2022-08-30T15:36:54+08:00",
"update_time" : "2022-08-30T15:36:54+08:00",
"l1" : null,
"l2" : null,
"l3" : null,
"summary_table_id" : null,
"approval_info" : null,
"new_biz" : null
},
"id" : "1014197025488699392",
"tables" : [ ],
"biz_type" : "DERIVATIVE_INDEX",
"parent_ids" : [ ],
"child_ids" : [ "1014195214723756032" ]
}],
"total" : 2,
"groups" : {
  "1014194851576721408" : {
    "dimension_groups" : [ {
      "group_id" : "1014194851576721408",
      "role" : null,
      "dimension_id" : "1014194851576721408",
      "hierarchies_id" : null,
      "ordinal" : 1,
      "group_name" : "testweidu830",
      "group_code" : "testweidu830",
      "biz_type" : "DIMENSION",
      "hierarchies" : null,
      "l1" : null,
      "l2" : null,
      "l3" : null,
      "l1_id" : null,
      "l2_id" : null,
      "l3_id" : null,
      "dw_type" : null,
      "id" : "1014197025585168384"
    } ],
    "ids" : [ "1014197025488699392" ]
  }
},
"links" : {
  "vo" : {
    "id" : "1014197025488699392",
    "name_en" : "zonghe8830",

```

```

"name_ch": "zonghe8830(testweidu830)",
"description": "",
"create_by": "abc",
"data_type": "STRING",
"l1_id": null,
"l2_id": null,
"l3_id": "767858832738234368",
"status": "DRAFT",
"atomic_index_id": "1014195214723756032",
"time_condition_id": null,
"time_field_id": null,
"time_field_name": null,
"common_conditions": [ ],
"dimension_groups": [ {
  "group_id": "1014194851576721408",
  "role": null,
  "dimension_id": "1014194851576721408",
  "hierarchies_id": null,
  "ordinal": 1,
  "group_name": "testweidu830",
  "group_code": "testweidu830",
  "biz_type": "DIMENSION",
  "hierarchies": null,
  "l1": null,
  "l2": null,
  "l3": null,
  "l1_id": null,
  "l2_id": null,
  "l3_id": null,
  "dw_type": null,
  "id": "1014197025585168384"
} ],
"monitor": null,
"atomic_index": {
  "id": "1014195214723756032",
  "name_en": "zonghe8830",
  "name_ch": "zonghe8830",
  "description": "",
  "create_by": null,
  "cal_exp": "sum(${1014195020200325120}) ",
  "cal_fn_ids": null,
  "l1_id": null,
  "l2_id": null,
  "l3_id": null,
  "table_id": "1014195020183547904",
  "tb_name": null,
  "dw_type": null,
  "field_ids": [ "1014195020200325120" ],
  "field_names": null,
  "status": "DRAFT",
  "biz_type": "FACT_LOGIC_TABLE",
  "create_time": "2022-09-02T09:56:31.536+08:00",
  "update_time": "2022-09-02T09:56:31.536+08:00",
  "l1": null,
  "l2": null,
  "l3": null,
  "approval_info": null,
  "new_biz": null
},
"time_condition_name": null,
"create_time": "2022-08-30T15:36:54+08:00",
"update_time": "2022-08-30T15:36:54+08:00",
"l1": null,
"l2": null,
"l3": null,
"summary_table_id": null,
"approval_info": null,
"new_biz": null
},

```

```
"id" : "1014197025488699392",  
"tables" : [],  
"biz_type" : "DERIVATIVE_INDEX",  
"parent_ids" : [],  
"child_ids" : [ "1014195214723756032" ]  
}  
}  
}
```

**状态码：400**

BadRequest

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user request is illegal."  
}
```

**状态码：401**

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

**状态码：403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## 状态码

状态码	描述
200	Success, 返回data数据是MetricRelationVOList。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.21 版本信息接口

### 8.21.1 查找版本信息

#### 功能介绍

通过名称、创建者、修改时间查找版本信息。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/design/versions

表 8-1311 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1312 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
biz_id	否	String	业务定义ID，ID字符串。
biz_type	否	String	按业务类型查询，可选业务类型有：ATOMIC_INDEX（原子指标）、DERIVATIVE_INDEX（衍生指标）、DIMENSION（维度）、TIME_CONDITION（时间限定）、DIMENSION_LOGIC_TABLE（维度表）、FACT_LOGIC_TABLE（事实表）、AGGREGATION_LOGIC_TABLE（汇总表）、TABLE_MODEL（关系建模表）、CODE_TABLE（码表）、STANDARD_ELEMENT（数据标准）、BIZ_METRIC（业务指标）、COMPOUND_METRIC（复合指标）、SUBJECT（主题）、ATOMIC_METRIC（原子指标（新））、DERIVED_METRIC（衍生指标（新））、COMPOSITE_METRIC（复合指标（新））。

参数	是否必选	参数类型	描述
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-1313 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1314 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回数据。

表 8-1315 data

参数	参数类型	描述
value	<b>value</b> object	返回的数据信息。

表 8-1316 value

参数	参数类型	描述
records	Array of <b>PublishVersionVO</b> objects	查询到的版本值对象 ( PublishVersionVO ) 数组。
total	Integer	符合搜索条件的记录总数。

表 8-1317 PublishVersionVO

参数	参数类型	描述
id	String	版本ID, ID字符串。
version_name	String	版本名称。
version_tag	String	版本标记, 只读。
description	String	版本描述。
biz_id	String	业务对象ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	业务详情，只读。
biz_info_vo	Object	业务对象。
effect_objs	String	影响信息，只读。
change_props	String	变化信息，只读。
sql_ddl	String	SQL脚本，只读。
physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
biz_metric	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
is_current_version	Boolean	是否为当前版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人，只读。

**状态码：400**

表 8-1318 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1319 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-1320 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**请求示例**

根据业务对象ID查找版本信息。

GET https://{endpoint}/v2/{project\_id}/design/versions?biz\_id=1193199235202428928

**响应示例**

**状态码：200**

Success，返回data数据是PublishVersionVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1196551286263267328",
        "version_name": "V1.0",
        "version_tag": "test_czh_001_2024-01-15_20-27-37_493",
        "description": null,
        "biz_id": "1193199235202428928",
        "biz_type": "BIZ_METRIC",
        "biz_info": null,
        "biz_info_vo": null,
        "effect_objs": null,
        "change_props": null,
        "sql_ddl": null,
        "physical_table": null,
        "technical_asset": null,
        "business_asset": "CREATE_SUCCESS",
        "meta_data_link": null,
        "data_quality": null,
        "dlf_task": null,
        "materialization": null,
        "publish_to_dlm": null,
        "biz_metric": "CREATE_SUCCESS",
        "summary_status": "SUMMARY_SUCCESS",
        "is_current_version": true,
        "create_time": "2024-01-15T20:27:37+08:00",
        "create_by": "test_uesr"
      }
    ]
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
```

```
"error_msg" : "The user request is illegal."  
}
```

**状态码: 401**

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

**状态码: 403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class SearchVersionsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        SearchVersionsRequest request = new SearchVersionsRequest();  
        try {  
            SearchVersionsResponse response = client.searchVersions(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchVersionsRequest()
        response = client.search_versions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.SearchVersionsRequest{}
response, err := client.SearchVersions(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是PublishVersionVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.21.2 比较版本信息

### 功能介绍

通过两个版本id，比较两者差异。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/design/versions/compare



表 8-1321 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1322 Query 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	实体ID数组，ID字符串。

## 请求参数

表 8-1323 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1324 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-1325 data

参数	参数类型	描述
value	<b>PublishVersionVO</b> object	CatalogVO信息。

表 8-1326 PublishVersionVO

参数	参数类型	描述
id	String	版本ID, ID字符串。
version_name	String	版本名称。
version_tag	String	版本标记, 只读。
description	String	版本描述。
biz_id	String	业务对象ID, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	业务详情，只读。
biz_info_vo	Object	业务对象。
effect_objs	String	影响信息，只读。
change_props	String	变化信息，只读。
sql_ddl	String	SQL脚本，只读。
physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dlf_task	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
materialization	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
publish_to_dlm	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
biz_metric	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>NO_NEED: 未同步</li> <li>CREATE_SUCCESS: 创建成功</li> <li>CREATE_FAILED: 创建失败</li> <li>UPDATE_SUCCESS: 更新成功</li> <li>UPDATE_FAILED: 更新失败</li> <li>SUMMARY_SUCCESS: 整体成功</li> <li>SUMMARY_FAILED: 整体失败</li> <li>RUNNING: 整体运行</li> <li>OFFLINE: 已下线</li> </ul>
is_current_version	Boolean	是否为当前版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人，只读。

**状态码：400**

表 8-1327 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1328 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。



**状态码：403**

**表 8-1329 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**请求示例**

比对版本ID为1295786824106741760与1301676931971940352之间的差异。

```
POST https://{endpoint}/v1/{project_id}/design/versions/compare?
ids=1295786824106741760&&ids=1301676931971940352
{ }
```

**响应示例**

**状态码：200**

Success, 返回data数据是 PublishVersionVO 详情。

```
{
  "data" : {
    "value" : {
      "id" : "1301676931971940352",
      "version_name" : "pd_test_2024-10-31_22-39-24_572",
      "version_tag" : "pd_test_2024-10-31_22-39-24_572",
      "description" : null,
      "biz_id" : "1295696904545603584",
      "biz_type" : "TABLE_MODEL",
      "biz_info" : "{ \"versionName\":null,\"versionTag\":null,\"versionId\":null,\"createBy
\\\": \"0833a573fe80d5401f6dc010a775569a\", \"updateBy\": \"0833a573fe80d5401f6dc010a775569a
\\\": \"1728959815000\", \"updateTime\": \"1730385564000\", \"l3Id\": \"1089989054919065600\", \"l2Id
\\\": null, \"alias\": null, \"encoding\": \"LE003444\", \"prefix\": \"LE\", \"codeNum\": null, \"extendInfo\": \"{
\\\": \"dirtyOutDatabase\": \"\", \"dirtyOutPrefix\": \"\", \"dirtyOutSuffix\": \"\", \"dirtyOutSwitch\":
\\\": \"false\", \"id\": \"1295696904545603584\", \"description\": \"无\", \"tenantId
\\\": \"0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70\", \"status
\\\": \"PUBLISHED\", \"tbName\": \"pd_test\", \"tbLogicName\": \"pd_test\", \"owner\": \"\", \"queueName
\\\": \"default\", \"schema\": null, \"dbName\": \"bi\", \"distribute\": null, \"distributeColumn\": null, \"compression
\\\": null, \"obsLocation\": null, \"preCombineField\": null, \"dwid
\\\": \"8259af52bd294f98920ebce75a426391\", \"dwName\": null, \"dwType\": \"DLI\", \"physicalTable
\\\": \"UPDATE_SUCCESS\", \"devPhysicalTable\": \"NO_NEED\", \"technicalAsset\": \"NO_NEED\", \"businessAsset
\\\": \"NO_NEED\", \"metaDataLink\": \"NO_NEED\", \"dataQuality\": \"UPDATE_FAILED\", \"dlfTask\": \"NO_NEED
\\\": \"SUMMARY_FAILED\", \"materialization\": null, \"publishToDtm\": null, \"envType
\\\": \"PROD_TYPE\", \"devVersion\": null, \"prodVersion\": \"1299388171930447872\", \"apild\": null, \"tbLogicGuid
\\\": \"33c30a2c-5f23-4bb5-a9c5-c1c0c8a7d609\", \"tbGuid\": null, \"tblid
\\\": \"NativeTable-8259af52bd294f98920ebce75a426391-pd_test\", \"partitionConf\": null, \"qualityOwner
\\\": null, \"qualityId\": \"1295789728175833089\", \"qualityIdExtend\": null, \"useRecentlyPartition
\\\": false, \"reversed\": null, \"dirtyOutSwitch\": false, \"dirtyOutDatabase\": \"\", \"dirtyOutPrefix
\\\": \"\", \"dirtyOutSuffix\": \"\", \"dlfTaskId\": null, \"modelId\": \"1295338804126908416\", \"model
\\\": null, \"selfDefinedFields\": null, \"scheduleTime\": null, \"parentTableId\": null, \"parentTableName
\\\": null, \"parentTableCode\": null, \"relatedLogicTableId\": \"1230888997367193600\", \"relatedLogicTableName
\\\": \"pd_test\", \"relatedLogicTableModelId\": \"1148908032844009472\", \"relatedLogicTableModelName
\\\": \"test_import\", \"logicTbName\": \"pd_test\", \"logicTbGuid\": \"33c30a2c-5f23-4bb5-a9c5-
c1c0c8a7d609\", \"logicTbId\": null, \"bizCatalogId\": \"1089989054919065600\", \"bizCatalogGuid
\\\": null, \"catalogPath\": null, \"secretType\": \"PUBLIC\", \"tableType\": \"MANAGED\", \"dataFormat\": \"Parquet
\\\": \"\", \"configs\": \"{}\", \"obsBucket\": null, \"approval\": null, \"sourceType\": 0, \"tags\": [], \"secrecyLevels
```

```

":null,\"attributes\":{[{"versionName":null,\"versionTag":null,\"versionId":null,\"createBy
\\":\"0833a573fe80d5401f6dc010a775569a\", \"updateBy\": \"0833a573fe80d5401f6dc010a775569a
\\\", \"createTime\": 1728959815000, \"updateTime\": 1730385565000, \"l3Id\": null, \"l2Id\": null, \"alias
\\\": \"\", \"encoding\": \"LP070256\", \"prefix\": \"LP\", \"codeNum\": null, \"id
\\\": \"1295696904545603585\", \"nameEn\": \"id\", \"nameCh\": \"id\", \"description\": \"\", \"obsLocation
\\\": null, \"dataType\": \"STRING\", \"isPrimaryKey\": true, \"isPartitionKey\": false, \"isForeignKey
\\\": false, \"extendField\": false, \"ordinal\": 1, \"tableModelId\": null, \"relatedLogicAttrId
\\\": \"1230888997920841728\", \"relatedLogicAttrName\": \"id\", \"relatedLogicAttrNameEn\": \"id
\\\", \"standRowId\": \"1294696939622932481\", \"standRowName\": \"1.1, avavvqav\", \"tags\":
[]}, {\"secrecyLevels\": [{\"versionName\": null, \"versionTag\": null, \"versionId\": null, \"createBy
\\\": \"ei_dayu_y00321344_01\", \"updateBy\": \"ei_dayu_y00321344_01\", \"createTime
\\\": 1730385564000, \"updateTime\": 1730385564000, \"l3Id\": null, \"l2Id\": null, \"alias\": null, \"encoding
\\\": null, \"prefix\": null, \"codeNum\": null, \"id\": \"20356\", \"secrecyLevelId\": \"1140704811168288768\", \"bizId
\\\": \"1295696904545603585\", \"secrecyLevelName\": \"zcy_admin_10\", \"uuiid\": \"\", \"slevel\": 0, \"description
\\\": null, \"bizType\": \"TABLE_MODEL\", \"tenantId\": \"0833a5737480d53b2f25c010dc1a7b88-workspace-
eeb055e69c624311b6b9cfee89a4ec70\", \"nameEn\": null, \"nameCh\": null, \"status\": null, \"lastL2Id
\\\": null}], \"dataTypeExtend\": \"\", \"domainType\": \"STRING\", \"qualityInfos\": [], \"standElementValues
\\\": null, \"selfDefinedFields\": [{\"fdNameCh\": \"隐私等级\", \"fdNameEn\": \"SecretLevel\", \"notNull
\\\": false, \"fdValue\": \"aa\", \"fdName\": \"隐私等级\", \"tenantId\": null, \"nameEn\": null, \"nameCh
\\\": null, \"createBy\": null, \"l3Id\": null, \"id\": null, \"status\": null}, {\"fdNameCh\": \"同义词\", \"fdNameEn
\\\": \"sameWord\", \"notNull\": false, \"fdValue\": \"bb\", \"fdName\": \"同义词\", \"tenantId\": null, \"nameEn
\\\": null, \"nameCh\": null, \"createBy\": null, \"l3Id\": null, \"id\": null, \"status\": null}], \"notNull\": true, \"code
\\\": \"LP070256\", \"bizCatalogId\": null, \"tenantId\": null, \"primaryKey\": true, \"partitionKey\": false, \"status
\\\": null, \"lastL2Id\": null}], \"relations\": [], \"reverseRelations\": [], \"mappings\": null, \"isPartition\": false, \"code
\\\": \"LE003444\", \"nameEn\": \"pd_test\", \"nameCh\": \"pd_test\", \"tagList\": [], \"secrecyLevelList\": [], \"lastL2Id
\\\": null}],
  \"biz_info_vo\" : {
    \"id\" : \"1295696904545603584\",
    \"model_id\" : \"1295338804126908416\",
    \"parent_table_id\" : null,
    \"parent_table_name\" : null,
    \"parent_table_code\" : null,
    \"related_logic_table_id\" : \"1230888997367193600\",
    \"related_logic_table_name\" : \"pd_test\",
    \"related_logic_table_model_id\" : \"1148908032844009472\",
    \"related_logic_table_model_name\" : \"test_import\",
    \"model\" : {
      \"id\" : \"1295338804126908416\",
      \"name\" : \"model_test\",
      \"description\" : \"\",
      \"is_physical\" : true,
      \"frequent\" : false,
      \"top\" : true,
      \"dw_type\" : null,
      \"create_time\" : 1.728874437E9,
      \"update_time\" : 1.728874437E9,
      \"create_by\" : \"ei_dayu_y00321344_01\",
      \"update_by\" : \"ei_dayu_y00321344_01\",
      \"type\" : \"THIRD_NF\",
      \"layer_id\" : \"1255490378069700608\",
      \"layer\" : {
        \"id\" : \"1255490378069700608\",
        \"level\" : 1,
        \"name\" : \"ODS\",
        \"type\" : \"THIRD_NF\",
        \"description\" : null,
        \"is_default\" : true,
        \"disabled_customized_field_ids\" : [ \"1254806436282773504\" ]
      },
      \"level\" : null,
      \"biz_catalog_ids\" : null,
      \"databases\" : null,
      \"table_model_prefix\" : \"\",
      \"dimension_prefix\" : null,
      \"is_default\" : false,
      \"has_permissions\" : null
    },
    \"data_format\" : \"Parquet\",
    \"obs_bucket\" : null,

```

```

"obs_location" : null,
"configs" : "{}",
"table_type" : "MANAGED",
"owner" : "",
"tb_name" : "pd_test",
"dw_id" : "8259af52bd294f98920ebce75a426391",
"db_name" : "bi",
"queue_name" : "default",
"schema" : null,
"extend_info" : "{\"dirtyOutDatabase\":\"\\\", \"dirtyOutPrefix\":\"\\\", \"dirtyOutSuffix
\\\": \"\\\", \"dirtyOutSwitch\":\"false\"}",
"tb_guid" : null,
"tb_id" : "NativeTable-8259af52bd294f98920ebce75a426391-pd_test",
"logic_tb_name" : "pd_test",
"logic_tb_guid" : "33c30a2c-5f23-4bb5-a9c5-c1c0c8a7d609",
"description" : "无",
"status" : "PUBLISHED",
"logic_tb_id" : null,
"biz_catalog_id" : "1089989054919065600",
"catalog_path" : "lx4/lxy_3",
"create_by" : "ei_dayu_y00321344_01",
"update_by" : "ei_dayu_y00321344_01",
"create_time" : 1.728959815E9,
"update_time" : 1.730385564E9,
"tags" : [ ],
"attributes" : [ {
  "id" : "1295696904545603585",
  "name_en" : "id",
  "name_ch" : "id",
  "description" : "",
  "obs_location" : null,
  "create_by" : "ei_dayu_y00321344_01",
  "update_by" : "ei_dayu_y00321344_01",
  "data_type" : "STRING",
  "domain_type" : "STRING",
  "data_type_extend" : "",
  "is_primary_key" : true,
  "is_partition_key" : false,
  "is_foreign_key" : false,
  "extend_field" : false,
  "not_null" : true,
  "ordinal" : 1,
  "table_model_id" : null,
  "create_time" : 1.728959815E9,
  "update_time" : 1.730385565E9,
  "tags" : [ ],
  "secrecy_levels" : [ {
    "id" : "20356",
    "secrecyLevel_id" : "1140704811168288768",
    "secrecyLevel_name" : "zcy_admin_10",
    "uuid" : "",
    "slevel" : 0,
    "description" : null,
    "biz_id" : "1295696904545603585",
    "biz_type" : "TABLE_MODEL",
    "create_by" : "ei_dayu_y00321344_01",
    "update_by" : "ei_dayu_y00321344_01",
    "create_time" : 1.730385564E9,
    "update_time" : 1.730385564E9
  } ],
  "stand_row_id" : "1294696939622932481",
  "stand_row_name" : "1.1, avavvqav",
  "quality_infos" : [ ],
  "alias" : "",
  "self_defined_fields" : [ {
    "fd_name_ch" : "隐私等级",
    "fd_name_en" : "SecretLevel",
    "not_null" : false,
    "fd_value" : "aa"
  } ]
} ]

```

```

}, {
  "fd_name_ch": "同义词",
  "fd_name_en": "sameWord",
  "not_null": false,
  "fd_value": "bb"
}],
"code": "LP070256",
"related_logic_attr_id": "1230888997920841728",
"related_logic_attr_name": "id",
"related_logic_attr_name_en": "id"
}],
"mappings": null,
"relations": [],
"dw_type": "DLI",
"dw_name": "dli",
"l1": "lx4",
"l2": "lx3",
"l3": null,
"l1_id": "1088851391579066368",
"l2_id": "1089989054919065600",
"l3_id": null,
"partition_conf": null,
"dlf_task_id": null,
"use_recently_partition": false,
"reversed": null,
"dirty_out_switch": false,
"dirty_out_database": "",
"dirty_out_prefix": "",
"dirty_out_suffix": "",
"quality_owner": null,
"quality_id": "1295789728175833089",
"quality_id_extend": null,
"distribute": null,
"distribute_column": null,
"compression": null,
"pre_combine_field": null,
"is_partition": false,
"physical_table": "UPDATE_SUCCESS",
"dev_physical_table": "NO_NEED",
"technical_asset": "NO_NEED",
"business_asset": "NO_NEED",
"meta_data_link": "NO_NEED",
"data_quality": "UPDATE_FAILED",
"summary_status": "SUMMARY_FAILED",
"dev_version": null,
"prod_version": "1299388171930447872",
"dev_version_name": null,
"prod_version_name": null,
"env_type": "PROD_TYPE",
"alias": null,
"self_defined_fields": null,
"code": "LE003444",
"has_related_physical_table": false,
"has_related_logic_table": false
},
"effect_objs": null,
"change_props": "基本信息:\n队列:CCdefault@CC\n数据库:CCbi@CC\n\n\n表字段:\n\nnid : 数据标准:CC1.1, avavqvav@CC.",
"sql_ddl": null,
"physical_table": "UPDATE_SUCCESS",
"technical_asset": "CREATE_SUCCESS",
"business_asset": "NO_NEED",
"meta_data_link": "UPDATE_SUCCESS",
"data_quality": "UPDATE_FAILED",
"dlf_task": "NO_NEED",
"materialization": null,
"publish_to_dlm": null,
"biz_metric": "NO_NEED",
"summary_status": "SUMMARY_FAILED",

```

```
"is_current_version" : false,  
"create_time" : "2024-10-31T22:39:25+08:00",  
"create_by" : "ei_dayu_y00321344_01"  
}  
}  
}
```

**状态码：400**

BadRequest

```
{  
"error_code" : "DS.60xx",  
"error_msg" : "The user request is illegal."  
}
```

**状态码：401**

Unauthorized

```
{  
"error_code" : "DS.60xx",  
"error_msg" : "User authentication failed."  
}
```

**状态码：403**

Forbidden

```
{  
"error_code" : "DS.60xx",  
"error_msg" : "The user does not have permission to call this API."  
}
```

## 状态码

状态码	描述
200	Success, 返回data数据是 PublishVersionVO 详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22 关系建模接口

### 8.22.1 查找表模型列表

#### 功能介绍

通过中英文名称、创建者、审核人、状态、修改时间分页查找关系建模中的表模型信息，包括逻辑实体、物理表和其属性。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/design/table-model

表 8-1330 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID</a> 和 <a href="#">账号ID</a> 。

表 8-1331 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
name_ch	否	String	按名称精确查询。
name_en	否	String	按编码精确查询。
create_by	否	String	按创建者查询。
approver	否	String	按审核人查询。
status	否	String	业务状态。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审批</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审批</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
sync_status	否	String	同步状态枚举。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 同步中</li> <li>• NO_NEED: 未同步</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> </ul>

参数	是否必选	参数类型	描述
sync_key	否	Array of strings	同步任务类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• BUSINESS_ASSET: 同步业务资产</li> <li>• DATA_QUALITY: 创建质量作业</li> <li>• TECHNICAL_ASSET: 同步技术资产</li> <li>• META_DATA_LINK: 资产关联</li> <li>• PHYSICAL_TABLE: 创建表（生产环境）</li> <li>• DEV_PHYSICAL_TABLE: 创建表（开发环境）</li> <li>• DLF_TASK: 创建数据开发作业</li> <li>• MATERIALIZATION: 数值落库（码表）</li> <li>• PUBLISH_TO_DLM: 发布数据服务API</li> <li>• SUMMARY_STATUS: 整体状态</li> </ul>
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。
model_id	是	String	所属关系建模的模型ID。

参数	是否必选	参数类型	描述
biz_catalog_id	否	String	所属的业务分层的ID。

## 请求参数

表 8-1332 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1333 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1334 data

参数	参数类型	描述
value	<b>value</b> object	value，统一的返回结果的外层数据结构。



表 8-1335 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>TableModelVO</b> objects	TableModelVO信息。

表 8-1336 TableModelVO

参数	参数类型	描述
id	String	编码，ID字符串。
model_id	String	所属关系建模的模型ID，ID字符串。
parent_table_id	String	父表ID，ID字符串。
parent_table_name	String	父表名称，只读。
parent_table_code	String	父表编码，只读。
related_logic_table_id	String	关联逻辑实体的ID，ID字符串。
related_logic_table_name	String	关联逻辑实体的名称。
related_logic_table_model_id	String	关联逻辑实体的模型ID，ID字符串。
related_logic_table_model_name	String	关联逻辑实体的模型名称。
model	<b>WorkspaceVO</b> object	模型信息。
data_format	String	数据格式。
obs_bucket	String	obs桶。
obs_location	String	外表路径
configs	String	其他配置。
table_type	String	表类型，只读。
owner	String	负责人。
tb_name	String	表名。
dw_id	String	数据连接ID。
db_name	String	数据库名。

参数	参数类型	描述
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。
extend_info	String	扩展信息。
tb_guid	String	表物化后的guid，只读。
tb_id	String	数据表ID，只读。
logic_tb_name	String	逻辑实体名。
logic_tb_guid	String	逻辑实体的guid，只读。
description	String	描述。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
logic_tb_id	String	逻辑实体的ID，ID字符串。
biz_catalog_id	String	归属的业务分类的id，ID字符串。
catalog_path	String	归属的业务分类的路径 { "l1Id": "", "l2Id": "", "l3Id": "" }。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
tags	Array of <a href="#">TagRecordVO</a> objects	表标签，只读。

参数	参数类型	描述
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息,包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理,只读。
attributes	Array of <b>TableModelAttributeVO</b> objects	表属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
relations	Array of <b>RelationVO</b> objects	关系。
dw_type	String	数据连接类型,对应表所在的数仓类型,取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称,只读,创建和更新时无需填写。
l1	String	主题域分组中文名,只读,创建和更新时无需填写。
l2	String	主题域中文名,只读,创建和更新时无需填写。
l3	String	业务对象中文名,只读,创建和更新时无需填写。
l1_id	String	主题域分组ID,只读,ID字符串。
l2_id	String	主题域ID,只读,创建和更新时无需填写。
l3_id	String	业务对象ID,只读,ID字符串。
partition_conf	String	分区表达式。
dlf_task_id	String	DLF作业ID。
use_recently_partition	Boolean	是否使用最新分区。
reversed	Boolean	是否是逆向的。
dirty_out_switch	Boolean	异常数据输出开关。

参数	参数类型	描述
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
quality_owner	String	质量责任人。
quality_id	String	质量ID, ID字符串。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值: <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash, 通过映射, 把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点 (DN) 中, 即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
is_partition	Boolean	是否分区表, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。

参数	参数类型	描述
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
code	String	编码
has_related_physical_table	Boolean	是否存在关联物理表。
has_related_logic_table	Boolean	是否存在关联逻辑实体。

表 8-1337 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>SDI: 贴源数据层</li> <li>DWI: 数据整合层</li> <li>DWR: 数据报告层</li> <li>DM: 数据集市层</li> </ul>

参数	参数类型	描述
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

表 8-1338 TagRecordVO

参数	参数类型	描述
id	String	编码，ID字符串。
tag_id	String	标签ID，ID字符串。
tag_name	String	标签名称。
biz_id	String	实体ID，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1339 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1340 TableModelAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
obs_location	String	obs路径，子路径。
create_by	String	创建人。
update_by	String	更新人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。

参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键。
extend_field	Boolean	是否继承的属性。
not_null	Boolean	是否不为空。
ordinal	Integer	序号。
table_model_id	String	所属关系建模的模型ID，ID字符串。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
tags	Array of <b>TagVO</b> objects	表标签，只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	String	逻辑属性编码
related_logic_attr_id	String	关联逻辑属性ID，ID字符串。
related_logic_attr_name	String	关联逻辑实体属性中文名称
related_logic_attr_name_en	String	关联逻辑实体属性英文名称

表 8-1341 TagVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	标签名。
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。

表 8-1342 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1343 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1344 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1345 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1346 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-1347 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1348 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-1349 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-1350 RelationVO

参数	参数类型	描述
id	String	编码, ID字符串。



参数	参数类型	描述
source_table_id	String	源表ID，ID字符串。
target_table_id	String	目标表ID，ID字符串。
name	String	关系名称。
source_table_name	String	源表名称。
target_table_name	String	目的表名称。
role	String	角色。
tenant_id	String	租户ID。
source_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
target_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
mappings	Array of <b>RelationMappingVO</b> objects	表属性信息，只读。

表 8-1351 RelationMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
relation_id	String	关系ID，ID字符串。
source_field_id	String	源字段ID，ID字符串。
target_field_id	String	目标字段ID，ID字符串。
source_field_name	String	源表名称。
target_field_name	String	目的表名称。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1352 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。

参数	参数类型	描述
not_null	Boolean	是否必填。
fd_value	String	属性值。

**状态码：400**

**表 8-1353 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-1354 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-1355 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

分页查询前十条关系建模表信息。

```
GET https://{endpoint}/v2/{project_id}/design/table-model?
offset=1&limit=10&model_id=1217123720355803136&no_attr=true
```

## 响应示例

### 状态码：200

Success, 返回data数据是TableModelVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1217123755214663682",
        "model_id": "1217123720355803136",
        "parent_table_id": null,
        "parent_table_name": null,
        "parent_table_code": null,
        "related_logic_table_id": "1217074041396031488",
        "related_logic_table_name": null,
        "related_logic_table_model_id": null,
        "related_logic_table_model_name": null,
        "model": null,
        "data_format": null,
        "obs_bucket": null,
        "obs_location": null,
        "configs": "{}",
        "table_type": "MYSQL_TABLE",
        "owner": null,
        "tb_name": "test_0312",
        "dw_id": "5013aa3f86df4335945131ce9d242b80",
        "db_name": null,
        "queue_name": null,
        "schema": null,
        "extend_info": "{\"columnTag\":\" \\\", \"columnSecrecyLevel\":\" \\\", \"tableTag\":\" \\\", \"dirtyOutSwitch\": \"false\", \"dirtyOutDatabase\":\" \\\", \"dirtyOutPrefix\":\" \\\", \"dirtyOutSuffix\":\" \\\"}",
        "tb_guid": null,
        "tb_id": null,
        "logic_tb_name": "test_0312",
        "logic_tb_guid": "dcd8753b-b864-4d1f-b431-1353009ceb1a",
        "description": "无aaaa",
        "status": "OFFLINE",
        "logic_tb_id": null,
        "biz_catalog_id": "1136041543187275776",
        "catalog_path": "ccbd",
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "create_time": "2024-03-12T14:55:16+08:00",
        "update_time": "2024-04-08T09:39:19+08:00",
        "tags": [ ],
        "approval_info": {
          "id": "1226828719683973120",
          "tenant_id": null,
          "name_ch": null,
          "name_en": null,
          "biz_id": "1217123755214663682",
          "biz_type": null,
          "biz_info": null,
          "biz_info_obj": null,
          "biz_version": 0,
          "biz_status": "OFFLINE",
          "approval_status": "APPROVED",
          "approval_type": null,
          "submit_time": "2024-04-08T09:39:20+08:00",
          "create_by": null,
          "l1": null,
          "l2": null,
          "l3": null,
          "approval_time": "2024-04-08T09:39:20+08:00",
          "approver": "test_uesr",
          "email": null,
        }
      } ]
    }
  }
}
```

```
    "msg": "Fast Approval OK.",
    "directory_path": null
  },
  "new_biz": null,
  "attributes": [],
  "mappings": null,
  "relations": [],
  "dw_type": "MYSQL",
  "dw_name": null,
  "l1": "ccbd",
  "l2": null,
  "l3": null,
  "l1_id": "1136041543187275776",
  "l2_id": null,
  "l3_id": null,
  "partition_conf": null,
  "dlf_task_id": null,
  "use_recently_partition": false,
  "reversed": null,
  "dirty_out_switch": false,
  "dirty_out_database": "",
  "dirty_out_prefix": "",
  "dirty_out_suffix": "",
  "quality_owner": null,
  "quality_id": null,
  "distribute": null,
  "distribute_column": null,
  "is_partition": false,
  "physical_table": "CREATE_FAILED",
  "dev_physical_table": "NO_NEED",
  "technical_asset": "CREATE_FAILED",
  "business_asset": "NO_NEED",
  "meta_data_link": "UPDATE_FAILED",
  "data_quality": "NO_NEED",
  "summary_status": "SUMMARY_FAILED",
  "dev_version": null,
  "prod_version": null,
  "dev_version_name": null,
  "prod_version_name": null,
  "env_type": "PROD_TYPE",
  "alias": null,
  "self_defined_fields": [ {
    "fd_name_ch": "aaa",
    "fd_name_en": "aaa",
    "not_null": null,
    "fd_value": ""
  } ],
  "code": "LE003441",
  "has_related_physical_table": false,
  "has_related_logic_table": true
}
}
```

**状态码: 400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码: 401**

Unauthorized

```
{
  "error_code": "DS.60xx",
```

```
"error_msg" : "User authentication failed."  
}
```

**状态码: 403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListTableModelsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListTableModelsRequest request = new ListTableModelsRequest();  
        try {  
            ListTableModelsResponse response = client.listTableModels(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListTableModelsRequest()
        response = client.list_table_models(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.ListTableModelsRequest{}  
response, err := client.ListTableModels(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是TableModelVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.2 创建表模型

### 功能介绍

在关系建模中创建一个表模型，包括逻辑实体和物理表。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/table-model

表 8-1356 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。



## 请求参数

表 8-1357 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1358 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
model_id	是	String	所属关系建模的模型ID，ID字符串。
parent_table_id	否	String	父表ID，ID字符串。
related_logic_table_id	否	String	关联逻辑实体的ID，ID字符串。
model	否	<a href="#">WorkspaceVO</a> object	模型信息。
data_format	否	String	数据格式。
obs_bucket	否	String	obs桶。
obs_location	否	String	外表路径
configs	否	String	其他配置。
owner	否	String	负责人。

参数	是否必选	参数类型	描述
tb_name	是	String	表名。
dw_id	否	String	数据连接ID。
db_name	否	String	数据库名。
queue_name	否	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	否	String	DWS类型需要。
extend_info	否	String	扩展信息。
logic_tb_name	是	String	逻辑实体名。
description	是	String	描述。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
logic_tb_id	否	String	逻辑实体的ID，ID字符串。
biz_catalog_id	否	String	归属的业务分类的id，ID字符串。
catalog_path	否	String	归属的业务分类的路径 { "l1Id": "", "l2Id": "", "l3Id": "" }。
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
attributes	是	Array of TableModelAttributeVO objects	表属性信息。
mappings	否	Array of TableMappingVO objects	表映射信息。

参数	是否必选	参数类型	描述
relations	否	Array of <b>RelationVO</b> objects	关系。
dw_type	是	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	否	String	数据连接名称，只读，创建和更新时无需填写。
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。
partition_conf	否	String	分区表达式。
dlf_task_id	否	String	DLF作业ID。
use_recently_partition	否	Boolean	是否使用最新分区。
reversed	否	Boolean	是否是逆向的。
dirty_out_switch	否	Boolean	异常数据输出开关。
dirty_out_database	否	String	异常数据输出库。
dirty_out_prefix	否	String	异常表前缀。
dirty_out_suffix	否	String	异常表后缀。
quality_owner	否	String	质量责任人。
quality_id	否	String	质量ID，ID字符串。
distribute	否	String	DISTRIBUTE BY [HASH(column)] REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行 Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>

参数	是否必选	参数类型	描述
distribute_column	否	String	DISTRIBUTE BY HASH column.
alias	否	String	别名。
self_defined_fields	否	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	否	String	编码

表 8-1359 WorkspaceVO

参数	是否必选	参数类型	描述
id	否	String	编号，ID字符串。
name	是	String	工作区名字。
description	否	String	描述。
is_physical	否	Boolean	是否为物理表。
frequent	否	Boolean	是否为常用。
top	否	Boolean	分层治理。
level	否	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	否	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_by	否	String	创建人。
update_by	否	String	更新人。
type	是	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>

参数	是否必选	参数类型	描述
biz_catalog_ids	否	String	关联的业务分层的ID列表 {“l1Ids”:[],“l2Ids”:[],“l3Ids”:[]}
databases	否	Array of strings	数据库名称数组。
table_model_prefix	否	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

表 8-1360 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。



表 8-1361 TableModelAttributeVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name_en	是	String	字段名。
name_ch	是	String	业务属性。
description	否	String	描述。
obs_location	否	String	obs路径，子路径。
create_by	否	String	创建人。
update_by	否	String	更新人。
data_type	是	String	字段类型。
data_type_ext_end	否	String	数据类型扩展字段。
is_primary_key	是	Boolean	是否主键。
is_partition_key	否	Boolean	是否分区键。
is_foreign_key	否	Boolean	是否外键。
extend_field	否	Boolean	是否继承的属性。
not_null	否	Boolean	是否不为空。
ordinal	否	Integer	序号。
table_model_id	否	String	所属关系建模的模型ID，ID字符串。
stand_row_id	否	String	关联的数据标准的ID，ID字符串。
alias	否	String	别名。
self_defined_fields	否	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	否	String	逻辑属性编码
related_logic_attr_id	否	String	关联逻辑属性ID，ID字符串。

表 8-1362 TableMappingVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name	是	String	名称。
description	否	String	描述。
target_table_id	否	String	目的表ID，ID字符串。
src_model_id	否	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	否	String	来源模型名称。
view_text	否	String	采集的视图来源，dws视图逆向使用。
target_table_name	否	String	目的表名称。
details	否	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	否	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	否	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	否	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	否	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-1363 TableMappingDetailVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
mapping_id	否	String	名称。
target_attr_id	否	String	目的字段ID，ID字符串。
target_attr_name	是	String	目的字段排序。
src_table_ids	否	String	源表ID。
src_attr_ids	否	String	源表字段ID。
remark	否	String	备注。
create_time	否	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	否	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	否	String	创建人。
update_by	否	String	更新人。
target_attr	否	Object	目标属性。

表 8-1364 MappingSourceTableVO

参数	是否必选	参数类型	描述
table1_id	是	String	表1ID，ID字符串。
table2_id	否	String	表2ID，ID字符串。
table1_name	是	String	表1名称。
table2_name	否	String	表2名称。

参数	是否必选	参数类型	描述
join_type	是	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	是	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1365 MappingJoinFieldVO

参数	是否必选	参数类型	描述
field1_id	是	String	属性1ID, ID字符串。
field2_id	是	String	属性2ID, ID字符串。
field1_name	是	String	属性1名称。
field2_name	是	String	属性2名称。

表 8-1366 MappingSourceFieldVO

参数	是否必选	参数类型	描述
target_field_id	否	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	是	String	目标字段编码。
field_ids	否	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	否	String	转换表达式。
field_names	否	Array of strings	来源字段名称列表。
changed	否	Boolean	字段是否发生变化。

表 8-1367 RelationVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
source_table_id	否	String	源表ID，ID字符串。
target_table_id	否	String	目标表ID，ID字符串。
name	是	String	关系名称。
source_table_name	否	String	源表名称。
target_table_name	否	String	目的表名称。
role	否	String	角色。
tenant_id	否	String	租户ID。
source_type	否	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>• ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>• ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>• ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>• ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>

参数	是否必选	参数类型	描述
target_type	否	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-1368 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

## 响应参数

状态码：200

表 8-1369 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1370 data

参数	参数类型	描述
value	<b>TableModelVO</b> object	value，统一的返回结果的外层数据结构。

表 8-1371 TableModelVO

参数	参数类型	描述
id	String	编码，ID字符串。
model_id	String	所属关系建模的模型ID，ID字符串。
parent_table_id	String	父表ID，ID字符串。
parent_table_name	String	父表名称，只读。
parent_table_code	String	父表编码，只读。
related_logic_table_id	String	关联逻辑实体的ID，ID字符串。
related_logic_table_name	String	关联逻辑实体的名称。
related_logic_table_model_id	String	关联逻辑实体的模型ID，ID字符串。
related_logic_table_model_name	String	关联逻辑实体的模型名称。
model	<b>WorkspaceVO</b> object	模型信息。
data_format	String	数据格式。
obs_bucket	String	obs桶。
obs_location	String	外表路径
configs	String	其他配置。
table_type	String	表类型，只读。
owner	String	负责人。
tb_name	String	表名。
dw_id	String	数据连接ID。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。

参数	参数类型	描述
schema	String	DWS类型需要。
extend_info	String	扩展信息。
tb_guid	String	表物化后的guid, 只读。
tb_id	String	数据表ID, 只读。
logic_tb_name	String	逻辑实体名。
logic_tb_guid	String	逻辑实体的guid, 只读。
description	String	描述。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
logic_tb_id	String	逻辑实体的ID, ID字符串。
biz_catalog_id	String	归属的业务分类的id, ID字符串。
catalog_path	String	归属的业务分类的路径 { "l1Id": "", "l2Id": "", "l3Id": "" }。
create_by	String	创建人, 只读。
update_by	String	更新人, 只读。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
tags	Array of <a href="#">TagRecordVO</a> objects	表标签, 只读。
approval_info	<a href="#">ApprovalVO</a> object	审批信息, 只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。



参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
attributes	Array of <b>TableModelAttributeVO</b> objects	表属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
relations	Array of <b>RelationVO</b> objects	关系。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，只读，ID字符串。
partition_conf	String	分区表达式。
dlf_task_id	String	DLF作业ID。
use_recently_partition	Boolean	是否使用最新分区。
reversed	Boolean	是否是逆向的。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。

参数	参数类型	描述
dirty_out_suffix	String	异常表后缀。
quality_owner	String	质量责任人。
quality_id	String	质量ID, ID字符串。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值: <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash, 通过映射, 把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点 (DN) 中, 即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
is_partition	Boolean	是否分区表, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>NO_NEED: 未同步</li> <li>CREATE_SUCCESS: 创建成功</li> <li>CREATE_FAILED: 创建失败</li> <li>UPDATE_SUCCESS: 更新成功</li> <li>UPDATE_FAILED: 更新失败</li> <li>SUMMARY_SUCCESS: 整体成功</li> <li>SUMMARY_FAILED: 整体失败</li> <li>RUNNING: 整体运行</li> <li>OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>NO_NEED: 未同步</li> <li>CREATE_SUCCESS: 创建成功</li> <li>CREATE_FAILED: 创建失败</li> <li>UPDATE_SUCCESS: 更新成功</li> <li>UPDATE_FAILED: 更新失败</li> <li>SUMMARY_SUCCESS: 整体成功</li> <li>SUMMARY_FAILED: 整体失败</li> <li>RUNNING: 整体运行</li> <li>OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>• INVALID_TYPE: 无效环境</li> <li>• DEV_TYPE: 开发环境</li> <li>• PROD_TYPE: 生产环境</li> <li>• DEV_PROD_TYPE: 开发生产环境</li> </ul>
alias	String	别名。

参数	参数类型	描述
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
code	String	编码
has_related_physical_table	Boolean	是否存在关联物理表。
has_related_logic_table	Boolean	是否存在关联逻辑实体。

表 8-1372 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

表 8-1373 TagRecordVO

参数	参数类型	描述
id	String	编码，ID字符串。
tag_id	String	标签ID，ID字符串。
tag_name	String	标签名称。
biz_id	String	实体ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1374 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1375 TableModelAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
obs_location	String	obs路径，子路径。
create_by	String	创建人。
update_by	String	更新人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。

参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键。
extend_field	Boolean	是否继承的属性。
not_null	Boolean	是否不为空。
ordinal	Integer	序号。
table_model_id	String	所属关系建模的模型ID，ID字符串。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
tags	Array of <b>TagVO</b> objects	表标签，只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	String	逻辑属性编码
related_logic_attr_id	String	关联逻辑属性ID，ID字符串。
related_logic_attr_name	String	关联逻辑实体属性中文名称
related_logic_attr_name_en	String	关联逻辑实体属性英文名称

表 8-1376 TagVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	标签名。
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。

表 8-1377 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1378 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-1379 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1380 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1381 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-1382 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1383 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-1384 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-1385 RelationVO

参数	参数类型	描述
id	String	编码, ID字符串。

参数	参数类型	描述
source_table_id	String	源表ID, ID字符串。
target_table_id	String	目标表ID, ID字符串。
name	String	关系名称。
source_table_name	String	源表名称。
target_table_name	String	目的表名称。
role	String	角色。
tenant_id	String	租户ID。
source_type	String	关系类型, 只读。 枚举值: <ul style="list-style-type: none"> <li>ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中至少有一条数据与之对应。</li> </ul>
target_type	String	关系类型, 只读。 枚举值: <ul style="list-style-type: none"> <li>ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
mappings	Array of <b>RelationMappingVO</b> objects	表属性信息，只读。

表 8-1386 RelationMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
relation_id	String	关系ID，ID字符串。
source_field_id	String	源字段ID，ID字符串。
target_field_id	String	目标字段ID，ID字符串。
source_field_name	String	源表名称。
target_field_name	String	目的表名称。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1387 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。



参数	参数类型	描述
not_null	Boolean	是否必填。
fd_value	String	属性值。

**状态码：400**

表 8-1388 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1389 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1390 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，创建物理表。

```
POST https://{endpoint}/v2/{project_id}/design/table-model
{
```

```

"attributes": [ {
  "id": 0,
  "code": "",
  "name_ch": "col_a",
  "name_en": "col_a",
  "alias": "",
  "description": "",
  "is_partition_key": false,
  "is_primary_key": false,
  "not_null": false,
  "ordinal": 1,
  "tags": [ {
    "name": "tag1"
  } ],
  "secrecy_levels": [ {
    "name": "secrecy_level_name"
  } ],
  "data_type": "STRING",
  "data_type_extend": "",
  "stand_row_id": null,
  "stand_row_name": null,
  "self_defined_fields": [ ]
}, {
  "id": 0,
  "code": "",
  "name_ch": "col_b",
  "name_en": "col_b",
  "alias": "",
  "description": "",
  "is_partition_key": false,
  "is_primary_key": false,
  "not_null": false,
  "ordinal": 2,
  "tags": [ ],
  "secrecy_levels": null,
  "data_type": "STRING",
  "data_type_extend": "",
  "stand_row_id": null,
  "stand_row_name": null,
  "self_defined_fields": [ ]
} ],
"mappings": [ {
  "name": "mapping_name"
} ],
"relations": [ {
  "name": "relation_name"
} ],
"model_id": "1208730797675311104",
"logic_tb_name": "test_create_table_model",
"tb_name": "test_create_table_model",
"code": null,
"alias": null,
"parent_table_id": null,
"related_logic_table_id": null,
"related_logic_table_model_id": null,
"configs": "{}",
"table_type": "HIVE_TABLE",
"data_format": "Parquet",
"obs_location": null,
"tags": [ {
  "biz_id": "1208730797675311105",
  "biz_type": "TABLE_MODEL",
  "tag_id": "1208730797675311105"
} ],
"biz_catalog_id": "958408897973161984",
"l1": "tesy",
"l2": "",
"l3": "",
"l1_id": "958408897973161984",

```

```

"l2_id" : "",
"l3_id" : "",
"catalog_path" : "{\\"l1d\\":\\"958408897973161984\\",\\"l2d\\":\\"\\",\\"l3d\\":\\"\\",\\"l4d\\":\\"\\",\\"l5d\\":\\"\\",\\"l6d\\":\\"\\",\\"l7d\\":\\"\\"}",
"dw_type" : "MRS_HIVE",
"dw_id" : "0fafdd43a342436e87e42d7217d26679",
"db_name" : null,
"queue_name" : null,
"schema" : null,
"owner" : "",
"description" : "无"
}

```

## 响应示例

**状态码：200**

Success, 返回data数据是TableModelVO。

```

{
  "data" : {
    "value" : {
      "id" : "1231921671028518912",
      "model_id" : "1208730797675311104",
      "parent_table_id" : null,
      "parent_table_name" : null,
      "parent_table_code" : null,
      "related_logic_table_id" : null,
      "related_logic_table_name" : null,
      "related_logic_table_model_id" : null,
      "related_logic_table_model_name" : null,
      "model" : null,
      "data_format" : "Parquet",
      "obs_bucket" : null,
      "obs_location" : null,
      "configs" : "{}",
      "table_type" : "HIVE_TABLE",
      "owner" : "",
      "tb_name" : "test_create_table_model",
      "dw_id" : "0fafdd43a342436e87e42d7217d26679",
      "db_name" : null,
      "queue_name" : null,
      "schema" : null,
      "extend_info" : "{\\"dirtyOutDatabase\\":\\"\\",\\"dirtyOutPrefix\\":\\"\\",\\"dirtyOutSuffix\\":\\"\\",\\"dirtyOutSwitch\\":\\"false\\"}",
      "tb_guid" : null,
      "tb_id" : null,
      "logic_tb_name" : "test_create_table_model",
      "logic_tb_guid" : null,
      "description" : "无",
      "status" : "DRAFT",
      "logic_tb_id" : null,
      "biz_catalog_id" : "958408897973161984",
      "catalog_path" : "{\\"l1d\\":\\"958408897973161984\\",\\"l2d\\":\\"\\",\\"l3d\\":\\"\\",\\"l4d\\":\\"\\",\\"l5d\\":\\"\\",\\"l6d\\":\\"\\",\\"l7d\\":\\"\\"}",
      "create_by" : "test_uesr",
      "update_by" : "test_uesr",
      "create_time" : "2024-04-22T10:56:54.371+08:00",
      "update_time" : "2024-04-22T10:56:54.371+08:00",
      "tags" : [],
      "approval_info" : null,
      "new_biz" : null,
      "attributes" : [ {
        "id" : "0",
        "name_en" : "col_a",
        "name_ch" : "col_a",
        "description" : "",
        "obs_location" : null,
        "create_by" : null,

```

```

"update_by" : null,
"data_type" : "STRING",
"domain_type" : "STRING",
"data_type_extend" : "",
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : false,
"extend_field" : false,
"not_null" : false,
"ordinal" : 1,
"table_model_id" : null,
"create_time" : "2024-04-22T10:56:54.371+08:00",
"update_time" : "2024-04-22T10:56:54.371+08:00",
"tags" : [ ],
"secrecy_levels" : null,
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ],
"code" : "",
"related_logic_attr_id" : null,
"related_logic_attr_name" : null,
"related_logic_attr_name_en" : null
}, {
"id" : "0",
"name_en" : "col_b",
"name_ch" : "col_b",
"description" : "",
"obs_location" : null,
"create_by" : null,
"update_by" : null,
"data_type" : "STRING",
"domain_type" : "STRING",
"data_type_extend" : "",
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : false,
"extend_field" : false,
"not_null" : false,
"ordinal" : 2,
"table_model_id" : null,
"create_time" : "2024-04-22T10:56:54.371+08:00",
"update_time" : "2024-04-22T10:56:54.371+08:00",
"tags" : [ ],
"secrecy_levels" : null,
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ],
"code" : "",
"related_logic_attr_id" : null,
"related_logic_attr_name" : null,
"related_logic_attr_name_en" : null
}],
"mappings" : [ ],
"relations" : [ ],
"dw_type" : "MRS_HIVE",
"dw_name" : null,
"l1" : null,
"l2" : null,
"l3" : null,
"l1_id" : null,
"l2_id" : "",
"l3_id" : "958408897973161984",
"partition_conf" : null,
"dlf_task_id" : null,
"use_recently_partition" : false,

```

```
"reversed" : false,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"quality_owner" : null,
"quality_id" : null,
"distribute" : null,
"distribute_column" : null,
"is_partition" : false,
"physical_table" : null,
"dev_physical_table" : null,
"technical_asset" : null,
"business_asset" : null,
"meta_data_link" : null,
"data_quality" : null,
"summary_status" : null,
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : null,
"alias" : null,
"self_defined_fields" : null,
"code" : null,
"has_related_physical_table" : false,
"has_related_logic_table" : false
}
}
```

**状态码：400**

## BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

## Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

## Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，创建物理表。

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateTableModelSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateTableModelRequest request = new CreateTableModelRequest();
        TableModelVO body = new TableModelVO();
        List<RelationVO> listbodyRelations = new ArrayList<>();
        listbodyRelations.add(
            new RelationVO()
                .withName("relation_name")
        );
        List<TableMappingVO> listbodyMappings = new ArrayList<>();
        listbodyMappings.add(
            new TableMappingVO()
                .withName("mapping_name")
        );
        List<SecrecyLevelVO> listAttributesSecrecyLevels = new ArrayList<>();
        listAttributesSecrecyLevels.add(
            new SecrecyLevelVO()
                .withName("secrecy_level_name")
        );
        List<TagVO> listAttributesTags = new ArrayList<>();
        listAttributesTags.add(
            new TagVO()
                .withName("tag1")
        );
        List<TableModelAttributeVO> listbodyAttributes = new ArrayList<>();
        listbodyAttributes.add(
            new TableModelAttributeVO()
                .withId("0")
                .withNameEn("col_a")
                .withNameCh("col_a")
                .withDescription("")
                .withDataType("STRING")
                .withDataTypeExtend("")
                .withIsPrimaryKey(false)
                .withIsPartitionKey(false)
                .withNotNull(false)
                .withOrdinal(1)
                .withTags(listAttributesTags)
        );
    }
}
```

```

        .withSecrecyLevels(listAttributesSecrecyLevels)
        .withAlias("")
        .withSelfDefinedFields()
        .withCode("")
    );
    listbodyAttributes.add(
        new TableModelAttributeVO()
            .withId("0")
            .withNameEn("col_b")
            .withNameCh("col_b")
            .withDescription("")
            .withDataType("STRING")
            .withDataTypeExtend("")
            .withIsPrimaryKey(false)
            .withIsPartitionKey(false)
            .withNotNull(false)
            .withOrdinal(2)
            .withTags()
            .withAlias("")
            .withSelfDefinedFields()
            .withCode("")
    );
    List<TagRecordVO> listbodyTags = new ArrayList<>();
    listbodyTags.add(
        new TagRecordVO()
            .withTagId("1208730797675311105")
            .withBizId("1208730797675311105")
            .withBizType(TagRecordVO.BizTypeEnum.fromValue("TABLE_MODEL"))
    );
    body.withL3Id("");
    body.withL2Id("");
    body.withL1Id("958408897973161984");
    body.withL3("");
    body.withL2("");
    body.withL1("tesy");
    body.withDwType("MRS_HIVE");
    body.withRelations(listbodyRelations);
    body.withMappings(listbodyMappings);
    body.withAttributes(listbodyAttributes);
    body.withTags(listbodyTags);

body.withCatalogPath("{\"l1Id\":\"958408897973161984\",\"l2Id\":\"\",\"l3Id\":\"\",\"l4Id\":\"\",\"l5Id\":\"\",\"l6Id\":\"\",\"l7Id\":\"\"}");
    );
    body.withBizCatalogId("958408897973161984");
    body.withDescription("无");
    body.withLogicTbName("test_create_table_model");
    body.withDwId("0fafdd43a342436e87e42d7217d26679");
    body.withTbName("test_create_table_model");
    body.withOwner("");
    body.withTableType("HIVE_TABLE");
    body.withConfigs("{}");
    body.withDataFormat("Parquet");
    body.withModelId("1208730797675311104");
    request.withBody(body);
    try {
        CreateTableModelResponse response = client.createTableModel(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}

```

```
}  
}
```

## Python

根据请求参数，创建物理表。

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateTableModelRequest()  
        listRelationsbody = [  
            RelationVO(  
                name="relation_name"  
            )  
        ]  
        listMappingsbody = [  
            TableMappingVO(  
                name="mapping_name"  
            )  
        ]  
        listSecrecyLevelsAttributes = [  
            SecrecyLevelVO(  
                name="secrecy_level_name"  
            )  
        ]  
        listTagsAttributes = [  
            TagVO(  
                name="tag1"  
            )  
        ]  
        listAttributesbody = [  
            TableModelAttributeVO(  
                id="0",  
                name_en="col_a",  
                name_ch="col_a",  
                description="",  
                data_type="STRING",  
                data_type_extend="",  
                is_primary_key=False,  
                is_partition_key=False,  
                not_null=False,  
                ordinal=1,  
                tags=listTagsAttributes,  
                secrecy_levels=listSecrecyLevelsAttributes,  
                alias="",
```



```

        code=""
    ),
    TableModelAttributeVO(
        id="0",
        name_en="col_b",
        name_ch="col_b",
        description="",
        data_type="STRING",
        data_type_extend="",
        is_primary_key=False,
        is_partition_key=False,
        not_null=False,
        ordinal=2,
        alias="",
        code=""
    )
]
listTagsbody = [
    TagRecordVO(
        tag_id="1208730797675311105",
        biz_id="1208730797675311105",
        biz_type="TABLE_MODEL"
    )
]
request.body = TableModelVO(
    l3_id="",
    l2_id="",
    l1_id="958408897973161984",
    l3="",
    l2="",
    l1="tesy",
    dw_type="MRS_HIVE",
    relations=listRelationsbody,
    mappings=listMappingsbody,
    attributes=listAttributesbody,
    tags=listTagsbody,
    catalog_path="{\"l1Id\":\"958408897973161984\",\"l2Id\":\"\",\"l3Id\":\"\",\"l4Id\":\"\",\"l5Id\":\"\",\"l6Id\":\"\",\"l7Id\":\"\"}",
    biz_catalog_id="958408897973161984",
    description="无",
    logic_tb_name="test_create_table_model",
    dw_id="0fafdd43a342436e87e42d7217d26679",
    tb_name="test_create_table_model",
    owner="",
    table_type="HIVE_TABLE",
    configs="{\"}",
    data_format="Parquet",
    model_id="1208730797675311104"
)
response = client.create_table_model(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

## Go

根据请求参数，创建物理表。

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateTableModelRequest{}
    var listRelationsbody = []model.RelationVo{
        {
            Name: "relation_name",
        },
    }
    var listMappingsbody = []model.TableMappingVo{
        {
            Name: "mapping_name",
        },
    }
    var listSecrecyLevelsAttributes = []model.SecrecyLevelVo{
        {
            Name: "secrecy_level_name",
        },
    }
    var listTagsAttributes = []model.TagVo{
        {
            Name: "tag1",
        },
    }
    idAttributes:= "0"
    descriptionAttributes:= ""
    dataTypeExtendAttributes:= ""
    isPartitionKeyAttributes:= false
    notNullAttributes:= false
    ordinalAttributes:= int32(1)
    aliasAttributes:= ""
    codeAttributes:= ""
    idAttributes1:= "0"
    descriptionAttributes1:= ""
    dataTypeExtendAttributes1:= ""
    isPartitionKeyAttributes1:= false
    notNullAttributes1:= false
    ordinalAttributes1:= int32(2)
    aliasAttributes1:= ""
    codeAttributes1:= ""
    var listAttributesbody = []model.TableModelAttributeVo{
        {
            Id: &idAttributes,
            NameEn: "col_a",
            NameCh: "col_a",
            Description: &descriptionAttributes,
            DataType: "STRING",
            DataTypeExtend: &dataTypeExtendAttributes,
            IsPrimaryKey: false,
        },
    }
}
```

```

        IsPartitionKey: &isPartitionKeyAttributes,
        NotNull: &notNullAttributes,
        Ordinal: &ordinalAttributes,
        Tags: &listTagsAttributes,
        SecrecyLevels: &listSecrecyLevelsAttributes,
        Alias: &aliasAttributes,
        Code: &codeAttributes,
    },
    {
        Id: &idAttributes1,
        NameEn: "col_b",
        NameCh: "col_b",
        Description: &descriptionAttributes1,
        DataType: "STRING",
        DataTypeExtend: &dataTypeExtendAttributes1,
        IsPrimaryKey: false,
        IsPartitionKey: &isPartitionKeyAttributes1,
        NotNull: &notNullAttributes1,
        Ordinal: &ordinalAttributes1,
        Alias: &aliasAttributes1,
        Code: &codeAttributes1,
    },
}
bizTypeTags:= model.GetTagRecordVoBizTypeEnum().TABLE_MODEL
var listTagsbody = []model.TagRecordVo{
    {
        TagId: "1208730797675311105",
        BizId: "1208730797675311105",
        BizType: &bizTypeTags,
    },
}
l3IdTableModelVo:= ""
l2IdTableModelVo:= ""
l1IdTableModelVo:= "958408897973161984"
l3TableModelVo:= ""
l2TableModelVo:= ""
l1TableModelVo:= "tesy"
catalogPathTableModelVo:=
"{\"l1Id\":\"958408897973161984\",\"l2Id\":\"\",\"l3Id\":\"\",\"l4Id\":\"\",\"l5Id\":\"\",\"l6Id\":\"\",\"l7Id\":\"\"}"
bizCatalogIdTableModelVo:= "958408897973161984"
dwIdTableModelVo:= "0fafdd43a342436e87e42d7217d26679"
ownerTableModelVo:= ""
tableTypeTableModelVo:= "HIVE_TABLE"
configsTableModelVo:= "{}"
dataFormatTableModelVo:= "Parquet"
request.Body = &model.TableModelVo{
    L3Id: &l3IdTableModelVo,
    L2Id: &l2IdTableModelVo,
    L1Id: &l1IdTableModelVo,
    L3: &l3TableModelVo,
    L2: &l2TableModelVo,
    L1: &l1TableModelVo,
    DwType: "MRS_HIVE",
    Relations: &listRelationsbody,
    Mappings: &listMappingsbody,
    Attributes: listAttributesbody,
    Tags: &listTagsbody,
    CatalogPath: &catalogPathTableModelVo,
    BizCatalogId: &bizCatalogIdTableModelVo,
    Description: "无",
    LogicTbName: "test_create_table_model",
    DwId: &dwIdTableModelVo,
    TbName: "test_create_table_model",
    Owner: &ownerTableModelVo,
    TableType: &tableTypeTableModelVo,
    Configs: &configsTableModelVo,
    DataFormat: &dataFormatTableModelVo,
    ModelId: "1208730797675311104",
}

```

```
response, err := client.CreateTableModel(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是TableModelVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.3 更新表模型

### 功能介绍

在关系建模中更新一个表模型及其属性，包括逻辑实体和物理表。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/table-model

表 8-1391 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-1392 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1393 请求 Body 参数

参数	是否必选	参数类型	描述
id	是	String	编码，ID字符串。
model_id	是	String	所属关系建模的模型ID，ID字符串。
parent_table_id	否	String	父表ID，ID字符串。
model	否	<a href="#">WorkspaceVO</a> object	模型信息。
data_format	否	String	数据格式。
obs_bucket	否	String	obs桶。
obs_location	否	String	obs路径。
configs	否	String	其他配置。
owner	否	String	责任人。
tb_name	是	String	表名。
dw_id	否	String	数据连接ID。

参数	是否必选	参数类型	描述
db_name	否	String	数据库名。
queue_name	否	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	否	String	DWS类型需要。
extend_info	否	String	扩展信息。
logic_tb_name	是	String	逻辑实体名。
description	是	String	描述。
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
logic_tb_id	否	String	逻辑实体的ID，ID字符串。
biz_catalog_id	否	String	归属的业务分类的ID，ID字符串。
catalog_path	否	String	归属的业务分类的路径，格式： {"l1id":"958408897973161984","l2id":"958408897973161985","l3id":"958408897973161986"}。
approval_info	否	ApprovalVO object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	否	BizVersionManagementVO object	业务版本管理，只读。
attributes	是	Array of TableModelAttributeVO objects	表属性信息。

参数	是否必选	参数类型	描述
mappings	否	Array of <b>TableMappingVO</b> objects	表映射信息。
relations	否	Array of <b>RelationVO</b> objects	关系。
dw_type	是	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	否	String	数据连接名称，只读，创建和更新时无需填写。
l2_id	否	String	主题域ID，只读，创建和更新时无需填写。
partition_conf	否	String	分区表达式
dlf_task_id	否	String	DLF作业ID。
use_recently_partition	否	Boolean	是否使用最新分区。
reversed	否	Boolean	是否是逆向的。
dirty_out_switch	否	Boolean	异常数据输出开关。
dirty_out_database	否	String	异常数据输出库。
dirty_out_prefix	否	String	异常表前缀。
dirty_out_suffix	否	String	异常表后缀。
quality_owner	否	String	质量责任人。
quality_id	否	String	质量ID，ID字符串。

参数	是否必选	参数类型	描述
distribute	否	String	DISTRIBUTE BY [HASH(column)  REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>HASH: 对指定的列进行 Hash，通过映射，把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	否	String	DISTRIBUTE BY HASH column.
alias	否	String	别名。
self_defined_fields	否	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
has_related_physical_table	否	Boolean	是否关联了物理表
has_related_logical_table	否	Boolean	是否关联了逻辑实体

表 8-1394 WorkspaceVO

参数	是否必选	参数类型	描述
id	否	String	编号，ID字符串。
name	是	String	工作区名字。
description	否	String	描述。
is_physical	否	Boolean	是否为物理表。
frequent	否	Boolean	是否为常用。
top	否	Boolean	分层治理。
level	否	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>SDI: 贴源数据层</li> <li>DWI: 数据整合层</li> <li>DWR: 数据报告层</li> <li>DM: 数据集市层</li> </ul>



参数	是否必选	参数类型	描述
dw_type	否	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_by	否	String	创建人。
update_by	否	String	更新人。
type	是	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	否	String	关联的业务分层的ID列表 {“l1Ids”:[],“l2Ids”:[],“l3Ids”:[]}
databases	否	Array of strings	数据库名称数组。
table_model_prefix	否	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

表 8-1395 ApprovalVO

参数	是否必选	参数类型	描述
id	否	String	审批单ID，ID字符串。
tenant_id	否	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	否	String	业务中文名。
name_en	否	String	业务英文名。
biz_id	否	String	业务ID，ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	否	String	序列化之后的业务详情，类型是string。
biz_info_obj	否	Object	业务详情，类型是object。
biz_version	否	Integer	业务版本。

参数	是否必选	参数类型	描述
biz_status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	否	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	否	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	否	String	提交时间。
create_by	否	String	创建者。
l1	否	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	否	String	主题域中文名，只读，创建和更新时无需填写。
l3	否	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	否	String	审核时间。
approver	否	String	审核人。
email	否	String	审核人邮箱。
msg	否	String	审核信息。
directory_path	否	String	目录树。

表 8-1396 BizVersionManageVO

参数	是否必选	参数类型	描述
id	否	String	ID信息, ID字符串。

参数	是否必选	参数类型	描述
biz_type	否	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标 (新)</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标 (新)</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标 (新)</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	否	String	业务ID，ID字符串。
biz_info	否	String	业务对象信息。

参数	是否必选	参数类型	描述
status	否	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>

表 8-1397 TableModelAttributeVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name_en	是	String	字段名。
name_ch	是	String	业务属性。
description	否	String	描述。
obs_location	否	String	obs路径，子路径。
create_by	否	String	创建人。
update_by	否	String	更新人。
data_type	是	String	字段类型。
data_type_extend	否	String	数据类型扩展字段。
is_primary_key	是	Boolean	是否主键。
is_partition_key	否	Boolean	是否分区键。
is_foreign_key	否	Boolean	是否外键。
extend_field	否	Boolean	是否继承的属性。
not_null	否	Boolean	是否不为空。
ordinal	否	Integer	序号。
table_model_id	否	String	所属关系建模的模型ID，ID字符串。



参数	是否必选	参数类型	描述
stand_row_id	否	String	关联的数据标准的ID，ID字符串。
alias	否	String	别名。
self_defined_fields	否	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	否	String	逻辑属性编码
related_logic_attr_id	否	String	关联逻辑属性ID，ID字符串。

表 8-1398 TableMappingVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
name	是	String	名称。
description	否	String	描述。
target_table_id	否	String	目的表ID，ID字符串。
src_model_id	否	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	否	String	来源模型名称。
view_text	否	String	采集的视图来源，dws视图逆向使用。
target_table_name	否	String	目的表名称。
details	否	Array of <b>TableMappingDetailVO</b> objects	详情。
source_tables	否	Array of <b>MappingSourceTableVO</b> objects	映射的表信息。
source_fields	否	Array of <b>MappingSourceFieldVO</b> objects	映射的字段信息。

参数	是否必选	参数类型	描述
create_time	否	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	否	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-1399 TableMappingDetailVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
mapping_id	否	String	名称。
target_attr_id	否	String	目的字段ID，ID字符串。
target_attr_name	是	String	目的字段排序。
src_table_ids	否	String	源表ID。
src_attr_ids	否	String	源表字段ID。
remark	否	String	备注。
create_time	否	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	否	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	否	String	创建人。
update_by	否	String	更新人。
target_attr	否	Object	目标属性。

表 8-1400 MappingSourceTableVO

参数	是否必选	参数类型	描述
table1_id	是	String	表1ID, ID字符串。
table2_id	否	String	表2ID, ID字符串。
table1_name	是	String	表1名称。
table2_name	否	String	表2名称。
join_type	是	String	关联类型。 枚举值: <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	是	Array of MappingJoinFieldVO objects	on条件。

表 8-1401 MappingJoinFieldVO

参数	是否必选	参数类型	描述
field1_id	是	String	属性1ID, ID字符串。
field2_id	是	String	属性2ID, ID字符串。
field1_name	是	String	属性1名称。
field2_name	是	String	属性2名称。

表 8-1402 MappingSourceFieldVO

参数	是否必选	参数类型	描述
target_field_id	否	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	是	String	目标字段编码。
field_ids	否	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	否	String	转换表达式。

参数	是否必选	参数类型	描述
field_names	否	Array of strings	来源字段名称列表。
changed	否	Boolean	字段是否发生变化。

表 8-1403 RelationVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。
source_table_id	否	String	源表ID，ID字符串。
target_table_id	否	String	目标表ID，ID字符串。
name	是	String	关系名称。
source_table_name	否	String	源表名称。
target_table_name	否	String	目的表名称。
role	否	String	角色。
tenant_id	否	String	租户ID。
source_type	否	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>• ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>• ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>• ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>• ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>

参数	是否必选	参数类型	描述
target_type	否	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	否	String	创建人。
update_by	否	String	更新人。

表 8-1404 SelfDefinedFieldVO

参数	是否必选	参数类型	描述
fd_name_ch	否	String	自定义项中文名。
fd_name_en	否	String	自定义项英文名。
not_null	否	Boolean	是否必填。
fd_value	否	String	属性值。

## 响应参数

状态码：200

表 8-1405 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1406 data

参数	参数类型	描述
value	<b>TableModelVO</b> object	value，统一的返回结果的外层数据结构。

表 8-1407 TableModelVO

参数	参数类型	描述
id	String	编码，ID字符串。
model_id	String	所属关系建模的模型ID，ID字符串。
parent_table_id	String	父表ID，ID字符串。
parent_table_name	String	父表名称，只读。
parent_table_code	String	父表编码，只读。
related_logic_table_id	String	关联逻辑实体的ID，ID字符串。
related_logic_table_name	String	关联逻辑实体的名称。
related_logic_table_model_id	String	关联逻辑实体的模型ID，ID字符串。
related_logic_table_model_name	String	关联逻辑实体的模型名称。
model	<b>WorkspaceVO</b> object	模型信息。
data_format	String	数据格式。
obs_bucket	String	obs桶。
obs_location	String	外表路径
configs	String	其他配置。
table_type	String	表类型，只读。
owner	String	负责人。
tb_name	String	表名。
dw_id	String	数据连接ID。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。

参数	参数类型	描述
schema	String	DWS类型需要。
extend_info	String	扩展信息。
tb_guid	String	表物化后的guid, 只读。
tb_id	String	数据表ID, 只读。
logic_tb_name	String	逻辑实体名。
logic_tb_guid	String	逻辑实体的guid, 只读。
description	String	描述。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
logic_tb_id	String	逻辑实体的ID, ID字符串。
biz_catalog_id	String	归属的业务分类的id, ID字符串。
catalog_path	String	归属的业务分类的路径 { "l1Id": "", "l2Id": "", "l3Id": "" }。
create_by	String	创建人, 只读。
update_by	String	更新人, 只读。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
tags	Array of <a href="#">TagRecordVO</a> objects	表标签, 只读。
approval_info	<a href="#">ApprovalVO</a> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。

参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
attributes	Array of <b>TableModelAttributeVO</b> objects	表属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
relations	Array of <b>RelationVO</b> objects	关系。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，只读，ID字符串。
partition_conf	String	分区表达式。
dlf_task_id	String	DLF作业ID。
use_recently_partition	Boolean	是否使用最新分区。
reversed	Boolean	是否是逆向的。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。



参数	参数类型	描述
dirty_out_suffix	String	异常表后缀。
quality_owner	String	质量责任人。
quality_id	String	质量ID, ID字符串。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值: <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash, 通过映射, 把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点 (DN) 中, 即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
is_partition	Boolean	是否分区表, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>NO_NEED: 未同步</li> <li>CREATE_SUCCESS: 创建成功</li> <li>CREATE_FAILED: 创建失败</li> <li>UPDATE_SUCCESS: 更新成功</li> <li>UPDATE_FAILED: 更新失败</li> <li>SUMMARY_SUCCESS: 整体成功</li> <li>SUMMARY_FAILED: 整体失败</li> <li>RUNNING: 整体运行</li> <li>OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>NO_NEED: 未同步</li> <li>CREATE_SUCCESS: 创建成功</li> <li>CREATE_FAILED: 创建失败</li> <li>UPDATE_SUCCESS: 更新成功</li> <li>UPDATE_FAILED: 更新失败</li> <li>SUMMARY_SUCCESS: 整体成功</li> <li>SUMMARY_FAILED: 整体失败</li> <li>RUNNING: 整体运行</li> <li>OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>• INVALID_TYPE: 无效环境</li> <li>• DEV_TYPE: 开发环境</li> <li>• PROD_TYPE: 生产环境</li> <li>• DEV_PROD_TYPE: 开发生产环境</li> </ul>
alias	String	别名。

参数	参数类型	描述
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
code	String	编码
has_related_physical_table	Boolean	是否存在关联物理表。
has_related_logic_table	Boolean	是否存在关联逻辑实体。

表 8-1408 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

表 8-1409 TagRecordVO

参数	参数类型	描述
id	String	编码，ID字符串。
tag_id	String	标签ID，ID字符串。
tag_name	String	标签名称。
biz_id	String	实体ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1410 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1411 TableModelAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
obs_location	String	obs路径，子路径。
create_by	String	创建人。
update_by	String	更新人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。

参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键。
extend_field	Boolean	是否继承的属性。
not_null	Boolean	是否不为空。
ordinal	Integer	序号。
table_model_id	String	所属关系建模的模型ID，ID字符串。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
tags	Array of <b>TagVO</b> objects	表标签，只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	String	逻辑属性编码
related_logic_attr_id	String	关联逻辑属性ID，ID字符串。
related_logic_attr_name	String	关联逻辑实体属性中文名称
related_logic_attr_name_en	String	关联逻辑实体属性英文名称

表 8-1412 TagVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	标签名。
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。

表 8-1413 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1414 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1415 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1416 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1417 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-1418 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>LEFT: 左外连接</li> <li>RIGHT: 右外连接</li> <li>INNER: 内连接</li> <li>FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1419 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-1420 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-1421 RelationVO

参数	参数类型	描述
id	String	编码, ID字符串。

参数	参数类型	描述
source_table_id	String	源表ID, ID字符串。
target_table_id	String	目标表ID, ID字符串。
name	String	关系名称。
source_table_name	String	源表名称。
target_table_name	String	目的表名称。
role	String	角色。
tenant_id	String	租户ID。
source_type	String	关系类型, 只读。 枚举值: <ul style="list-style-type: none"> <li>ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中至少有一条数据与之对应。</li> </ul>
target_type	String	关系类型, 只读。 枚举值: <ul style="list-style-type: none"> <li>ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
mappings	Array of <b>RelationMapping VO</b> objects	表属性信息，只读。

表 8-1422 RelationMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
relation_id	String	关系ID，ID字符串。
source_field_id	String	源字段ID，ID字符串。
target_field_id	String	目标字段ID，ID字符串。
source_field_name	String	源表名称。
target_field_name	String	目的表名称。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1423 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。

参数	参数类型	描述
not_null	Boolean	是否必填。
fd_value	String	属性值。

**状态码：400**

表 8-1424 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1425 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1426 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，更新物理表。

```
PUT https://{endpoint}/v2/{project_id}/design/table-model
{
```



```

"id": "1231932180633112576",
"model_id": "1208730797675311104",
"parent_table_id": null,
"parent_table_name": null,
"parent_table_code": null,
"model": {
  "id": "1208730797675311104",
  "name": "xingneng2",
  "description": "",
  "is_physical": true,
  "frequent": false,
  "top": false,
  "level": null,
  "dw_type": "MRS_HIVE",
  "create_time": "2024-02-18T11:04:39+08:00",
  "update_time": "2024-02-18T11:04:39+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "type": "THIRD_NF",
  "biz_catalog_ids": null,
  "databases": null,
  "table_model_prefix": ""
},
"data_format": "Parquet",
"obs_bucket": null,
"obs_location": null,
"configs": "{}",
"table_type": "HIVE_TABLE",
"owner": "",
"tb_name": "test_update_table_model",
"dw_id": "0fafdd43a342436e87e42d7217d26679",
"db_name": null,
"queue_name": null,
"schema": null,
"extend_info": "{\"dirtyOutDatabase\":\"\",\"dirtyOutPrefix\":\"\",\"dirtyOutSuffix\":\"\",\"dirtyOutSwitch\":\"false\"}",
"tb_guid": null,
"tb_id": null,
"logic_tb_name": "test_update_table_model",
"logic_tb_guid": null,
"description": "无",
"status": "DRAFT",
"logic_tb_id": null,
"biz_catalog_id": "958408897973161984",
"catalog_path": "{\"l1Id\":\"958408897973161984\",\"l2Id\":\"\",\"l3Id\":\"\",\"l4Id\":\"\",\"l5Id\":\"\",\"l6Id\":\"\",\"l7Id\":\"\"}",
"create_by": "test_uesr",
"update_by": "test_uesr",
"create_time": "2024-04-22T11:38:40+08:00",
"update_time": "2024-04-22T11:38:40+08:00",
"tags": [ {
  "name": "tag1"
} ],
"approval_info": null,
"new_biz": null,
"attributes": [ {
  "id": "1231932180754747392",
  "name_en": "col_a",
  "name_ch": "col_a",
  "description": "",
  "obs_location": null,
  "create_by": null,
  "update_by": null,
  "data_type": "STRING",
  "domain_type": "STRING",
  "data_type_extend": "",
  "is_primary_key": false,
  "is_partition_key": false,
  "is_foreign_key": false,

```

```

"extend_field" : false,
"not_null" : false,
"ordinal" : 1,
"table_model_id" : null,
"create_time" : "2024-04-22T11:38:40+08:00",
"update_time" : "2024-04-22T11:38:40+08:00",
"tags" : [ {
  "name" : "tag1"
} ],
"secrecy_levels" : [ {
  "name" : "secrecy_level_name"
} ],
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : [ {
  "data_quality_id" : "1231932180754747392"
} ],
"alias" : "",
"self_defined_fields" : [ {
  "fd_name_ch" : "芬",
  "fd_name_en" : "SDSDFD",
  "not_null" : false,
  "fd_value" : ""
}, {
  "fd_name_ch" : "test",
  "fd_name_en" : "test",
  "not_null" : false,
  "fd_value" : ""
} ],
"code" : "",
"related_logic_attr_id" : null,
"related_logic_attr_name" : null,
"related_logic_attr_name_en" : null
} ],
"mappings" : [ {
  "name" : "mapping_name"
} ],
"relations" : [ {
  "name" : "relation_name"
} ],
"dw_type" : "MRS_HIVE",
"dw_name" : "MRS_Hive",
"l1" : "tesy",
"l2" : "",
"l3" : null,
"l1_id" : "958408897973161984",
"l2_id" : "",
"l3_id" : null,
"partition_conf" : null,
"dlf_task_id" : null,
"use_recently_partition" : false,
"reversed" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"quality_owner" : null,
"quality_id" : null,
"distribute" : null,
"distribute_column" : null,
"is_partition" : false,
"physical_table" : "NO_NEED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "NO_NEED",
"business_asset" : "NO_NEED",
"meta_data_link" : "NO_NEED",
"data_quality" : "NO_NEED",
"summary_status" : "NO_NEED",
"dev_version" : null,

```

```
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "INVALID_TYPE",
"alias" : null,
"self_defined_fields" : null,
"has_related_physical_table" : false,
"has_related_logic_table" : false
}
```

## 响应示例

**状态码: 200**

Success, 返回data数据是TableModelVO。

```
{
  "data" : {
    "value" : {
      "id" : "1012314146584301568",
      "model_id" : "873242752312385536",
      "parent_table_id" : null,
      "parent_table_name" : null,
      "parent_table_code" : null,
      "model" : null,
      "data_format" : "Parquet",
      "obs_bucket" : null,
      "obs_location" : null,
      "configs" : "{}",
      "table_type" : "MANAGED",
      "owner" : "",
      "tb_name" : "staff_info",
      "dw_id" : "2c9080f07b03db8d017b198a4a20002f",
      "db_name" : "ctz_test",
      "queue_name" : "default",
      "schema" : "",
      "extend_info" : "{\"dirtyOutDatabase\":\"\",\"dirtyOutPrefix\":\"\",\"dirtyOutSuffix\":\"\",\"dirtyOutSwitch\":\"false\"}",
      "tb_guid" : null,
      "tb_id" : "NativeTable-2c9080f07b03db8d017b198a4a20002f-ctz_test-staff_info",
      "logic_tb_name" : "staff_info",
      "logic_tb_guid" : "1ab1a58b-9412-4901-bb80-4ccaaa0580e4",
      "description" : "无",
      "status" : "PUBLISHED",
      "logic_tb_id" : null,
      "biz_catalog_id" : "1012306178602041344",
      "catalog_path" : "工厂/人员管理/人员档案",
      "create_by" : "abc",
      "update_by" : "abc",
      "create_time" : "2022-08-25T10:55:00+08:00",
      "update_time" : "2022-08-25T10:55:45+08:00",
      "tags" : [],
      "approval_info" : null,
      "new_biz" : null,
      "attributes" : [ {
        "id" : "1012314146626244608",
        "name_en" : "id",
        "name_ch" : "主键",
        "description" : "",
        "obs_location" : null,
        "create_by" : "abc",
        "update_by" : "abc",
        "data_type" : "STRING",
        "domain_type" : "STRING",
        "data_type_extend" : "",
        "is_primary_key" : true,
        "is_partition_key" : false,
        "is_foreign_key" : false,
        "extend_field" : false,
      } ]
    }
  }
}
```

```

"not_null" : true,
"ordinal" : 1,
"table_model_id" : null,
"create_time" : "2022-08-25T10:55:00+08:00",
"update_time" : "2022-08-25T10:55:00+08:00",
"tags" : [ ],
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}, {
"id" : "1012314146626244609",
"name_en" : "name",
"name_ch" : "名称",
"description" : "",
"obs_location" : null,
"create_by" : "abc",
"update_by" : "abc",
"data_type" : "STRING",
"domain_type" : "STRING",
"data_type_extend" : "",
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : false,
"extend_field" : false,
"not_null" : false,
"ordinal" : 2,
"table_model_id" : null,
"create_time" : "2022-08-25T10:55:00+08:00",
"update_time" : "2022-08-25T10:55:00+08:00",
"tags" : [ ],
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}, {
"id" : "1012314146592690176",
"name_en" : "age",
"name_ch" : "年龄",
"description" : "",
"obs_location" : null,
"create_by" : "abc",
"update_by" : "abc",
"data_type" : "STRING",
"domain_type" : "STRING",
"data_type_extend" : "",
"is_primary_key" : false,
"is_partition_key" : false,
"is_foreign_key" : false,
"extend_field" : false,
"not_null" : false,
"ordinal" : 3,
"table_model_id" : null,
"create_time" : "2022-08-25T10:55:00+08:00",
"update_time" : "2022-08-25T10:55:00+08:00",
"tags" : [ ],
"stand_row_id" : null,
"stand_row_name" : null,
"quality_infos" : null,
"alias" : "",
"self_defined_fields" : [ ]
}],
"mappings" : null,
"relations" : [ ],
"dw_type" : "DWS",
"dw_name" : null,
"l1" : "工厂",

```

```
"l2": "人员管理",
"l3": "人员档案",
"l1_id": "1012305751919689728",
"l2_id": "1012306048792526848",
"l3_id": "1012306178602041344",
"partition_conf": null,
"dlf_task_id": null,
"use_recently_partition": false,
"reversed": null,
"dirty_out_switch": false,
"dirty_out_database": "",
"dirty_out_prefix": "",
"dirty_out_suffix": "",
"quality_owner": null,
"quality_id": "1012314332484304897",
"distribute": null,
"distribute_column": null,
"is_partition": false,
"physical_table": "CREATE_SUCCESS",
"technical_asset": "CREATE_FAILED",
"business_asset": "UPDATE_SUCCESS",
"meta_data_link": "UPDATE_FAILED",
"data_quality": "CREATE_SUCCESS",
"summary_status": "SUMMARY_FAILED",
"alias": null,
"self_defined_fields": null
}
}
}
```

**状态码: 400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码: 401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码: 403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，更新物理表。

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.time.OffsetDateTime;
import java.time.format.DateTimeFormatter;
import java.util.List;
import java.util.ArrayList;

public class UpdateTableModelSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTableModelRequest request = new UpdateTableModelRequest();
        TableModelUpdateVO body = new TableModelUpdateVO();
        List<RelationVO> listbodyRelations = new ArrayList<>();
        listbodyRelations.add(
            new RelationVO()
                .withName("relation_name")
        );
        List<TableMappingVO> listbodyMappings = new ArrayList<>();
        listbodyMappings.add(
            new TableMappingVO()
                .withName("mapping_name")
        );
        List<SelfDefinedFieldVO> listAttributesSelfDefinedFields = new ArrayList<>();
        listAttributesSelfDefinedFields.add(
            new SelfDefinedFieldVO()
                .withFdNameCh("芬")
                .withFdNameEn("SDSDFD")
                .withNotNull(false)
                .withFdValue("")
        );
        listAttributesSelfDefinedFields.add(
            new SelfDefinedFieldVO()
                .withFdNameCh("test")
                .withFdNameEn("test")
                .withNotNull(false)
                .withFdValue("")
        );
        List<QualityInfoVO> listAttributesQualityInfos = new ArrayList<>();
        listAttributesQualityInfos.add(
            new QualityInfoVO()
                .withDataQualityId("1231932180754747392")
        );
        List<SecrecyLevelVO> listAttributesSecrecyLevels = new ArrayList<>();
        listAttributesSecrecyLevels.add(
```

```
new SecrecyLevelVO()
    .withName("secrecy_level_name")
);
List<TagVO> listAttributesTags = new ArrayList<>();
listAttributesTags.add(
    new TagVO()
        .withName("tag1")
);
List<TableModelAttributeVO> listbodyAttributes = new ArrayList<>();
listbodyAttributes.add(
    new TableModelAttributeVO()
        .withId("1231932180754747392")
        .withNameEn("col_a")
        .withNameCh("col_a")
        .withDescription("")
        .withDataType("STRING")
        .withDomainType(TableModelAttributeVO.DomainTypeEnum.fromValue("STRING"))
        .withDataTypeExtend("")
        .withIsPrimaryKey(false)
        .withIsPartitionKey(false)
        .withIsForeignKey(false)
        .withExtendField(false)
        .withNotNull(false)
        .withOrdinal(1)
        .withCreateTime(OffsetDateTime.parse("2024-04-22T11:38:40+08:00",
DateTimeFormatter.ISO_OFFSET_DATE_TIME))
        .withUpdateTime(OffsetDateTime.parse("2024-04-22T11:38:40+08:00",
DateTimeFormatter.ISO_OFFSET_DATE_TIME))
        .withTags(listAttributesTags)
        .withSecrecyLevels(listAttributesSecrecyLevels)
        .withQualityInfos(listAttributesQualityInfos)
        .withAlias("")
        .withSelfDefinedFields(listAttributesSelfDefinedFields)
        .withCode("")
);
List<TagVO> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new TagVO()
        .withName("tag1")
);
WorkspaceVO modelbody = new WorkspaceVO();
modelbody.withId("1208730797675311104")
    .withName("xingneng2")
    .withDescription("")
    .withIsPhysical(true)
    .withFrequent(false)
    .withTop(false)
    .withDwType("MRS_HIVE")
    .withCreateTime(OffsetDateTime.parse("2024-02-18T11:04:39+08:00",
DateTimeFormatter.ISO_OFFSET_DATE_TIME))
    .withUpdateTime(OffsetDateTime.parse("2024-02-18T11:04:39+08:00",
DateTimeFormatter.ISO_OFFSET_DATE_TIME))
    .withCreateBy("test_uesr")
    .withUpdateBy("test_uesr")
    .withType(WorkspaceVO.TypeEnum.fromValue("THIRD_NF"))
    .withTableModelPrefix("");
body.withHasRelatedLogicTable(false);
body.withHasRelatedPhysicalTable(false);
body.withEnvType(TableModelUpdateVO.EnvTypeEnum.fromValue("INVALID_TYPE"));
body.withSummaryStatus(TableModelUpdateVO.SummaryStatusEnum.fromValue("NO_NEED"));
body.withDataQuality(TableModelUpdateVO.DataQualityEnum.fromValue("NO_NEED"));
body.withMetaDataLink(TableModelUpdateVO.MetaDataLinkEnum.fromValue("NO_NEED"));
body.withBusinessAsset(TableModelUpdateVO.BusinessAssetEnum.fromValue("NO_NEED"));
body.withTechnicalAsset(TableModelUpdateVO.TechnicalAssetEnum.fromValue("NO_NEED"));
body.withDevPhysicalTable(TableModelUpdateVO.DevPhysicalTableEnum.fromValue("NO_NEED"));
body.withPhysicalTable(TableModelUpdateVO.PhysicalTableEnum.fromValue("NO_NEED"));
body.withIsPartition(false);
body.withDirtyOutSuffix("");
body.withDirtyOutPrefix("");
```

```
body.withDirtyOutDatabase("");
body.withDirtyOutSwitch(false);
body.withUseRecentlyPartition(false);
body.withL2Id("");
body.withL1Id("958408897973161984");
body.withL2("");
body.withL1("tesy");
body.withDwName("MRS_Hive");
body.withDwType("MRS_HIVE");
body.withRelations(listbodyRelations);
body.withMappings(listbodyMappings);
body.withAttributes(listbodyAttributes);
body.withTags(listbodyTags);
body.withUpdateTime(OffsetDateTime.parse("2024-04-22T11:38:40+08:00",
DateTimeFormatter.ISO_OFFSET_DATE_TIME));
body.withCreateTime(OffsetDateTime.parse("2024-04-22T11:38:40+08:00",
DateTimeFormatter.ISO_OFFSET_DATE_TIME));
body.withUpdateBy("test_uesr");
body.withCreateBy("test_uesr");

body.withCatalogPath("{\"l1Id\":\"958408897973161984\",\"l2Id\":\"\",\"l3Id\":\"\",\"l4Id\":\"\",\"l5Id\":\"\",\"l6Id\":\"\",\"l7Id\":\"\"}");
body.withBizCatalogId("958408897973161984");
body.withStatus(TableModelUpdateVO.StatusEnum.fromValue("DRAFT"));
body.withDescription("无");
body.withLogicTbName("test_update_table_model");

body.withExtendInfo("{\"dirtyOutDatabase\":\"\",\"dirtyOutPrefix\":\"\",\"dirtyOutSuffix\":\"\",\"dirtyOutSwitch\":\"false\"}");
body.withDwId("0fafdd43a342436e87e42d7217d26679");
body.withTbName("test_update_table_model");
body.withOwner("");
body.withTableType("HIVE_TABLE");
body.withConfigs("{}");
body.withDataFormat("Parquet");
body.withModel(modelbody);
body.withModelId("1208730797675311104");
body.withId("1231932180633112576");
request.withBody(body);
try {
    UpdateTableModelResponse response = client.updateTableModel(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据请求参数，更新物理表。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
```



```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateTableModelRequest()
    listRelationsbody = [
        RelationVO(
            name="relation_name"
        )
    ]
    listMappingsbody = [
        TableMappingVO(
            name="mapping_name"
        )
    ]
    listSelfDefinedFieldsAttributes = [
        SelfDefinedFieldVO(
            fd_name_ch="芬",
            fd_name_en="SDSDFD",
            not_null=False,
            fd_value=""
        ),
        SelfDefinedFieldVO(
            fd_name_ch="test",
            fd_name_en="test",
            not_null=False,
            fd_value=""
        )
    ]
    listQualityInfosAttributes = [
        QualityInfoVO(
            data_quality_id="1231932180754747392"
        )
    ]
    listSecrecyLevelsAttributes = [
        SecrecyLevelVO(
            name="secrecy_level_name"
        )
    ]
    listTagsAttributes = [
        TagVO(
            name="tag1"
        )
    ]
    listAttributesbody = [
        TableModelAttributeVO(
            id="1231932180754747392",
            name_en="col_a",
            name_ch="col_a",
            description="",
            data_type="STRING",
            domain_type="STRING",
            data_type_extend="",
            is_primary_key=False,
            is_partition_key=False,
            is_foreign_key=False,
```

```
        extend_field=False,
        not_null=False,
        ordinal=1,
        create_time="2024-04-22T11:38:40+08:00",
        update_time="2024-04-22T11:38:40+08:00",
        tags=listTagsAttributes,
        secrecy_levels=listSecrecyLevelsAttributes,
        quality_infos=listQualityInfosAttributes,
        alias="",
        self_defined_fields=listSelfDefinedFieldsAttributes,
        code=""
    )
]
listTagsbody = [
    TagVO(
        name="tag1"
    )
]
modelbody = WorkspaceVO(
    id="1208730797675311104",
    name="xingneng2",
    description="",
    is_physical=True,
    frequent=False,
    top=False,
    dw_type="MRS_HIVE",
    create_time="2024-02-18T11:04:39+08:00",
    update_time="2024-02-18T11:04:39+08:00",
    create_by="test_uesr",
    update_by="test_uesr",
    type="THIRD_NF",
    table_model_prefix=""
)
request.body = TableModelUpdateVO(
    has_related_logic_table=False,
    has_related_physical_table=False,
    env_type="INVALID_TYPE",
    summary_status="NO_NEED",
    data_quality="NO_NEED",
    meta_data_link="NO_NEED",
    business_asset="NO_NEED",
    technical_asset="NO_NEED",
    dev_physical_table="NO_NEED",
    physical_table="NO_NEED",
    is_partition=False,
    dirty_out_suffix="",
    dirty_out_prefix="",
    dirty_out_database="",
    dirty_out_switch=False,
    use_recently_partition=False,
    l2_id="",
    l1_id="958408897973161984",
    l2="",
    l1="tesy",
    dw_name="MRS_Hive",
    dw_type="MRS_HIVE",
    relations=listRelationsbody,
    mappings=listMappingsbody,
    attributes=listAttributesbody,
    tags=listTagsbody,
    update_time="2024-04-22T11:38:40+08:00",
    create_time="2024-04-22T11:38:40+08:00",
    update_by="test_uesr",
    create_by="test_uesr",
    catalog_path="{\"l1Id\":\"958408897973161984\",\"l2Id\":\"\",\"l3Id\":\"\",\"l4Id\":\"\",\"l5Id\":\"\",\"l6Id\":\"\",\"l7Id\":\"\"}",
    biz_catalog_id="958408897973161984",
    status="DRAFT",
    description="无",
    logic_tb_name="test_update_table_model",
```

```
extend_info="{\"dirtyOutDatabase\":\"\", \"dirtyOutPrefix\":\"\", \"dirtyOutSuffix\":\"\", \"dirtyOutSwitch\":\"false\"}\",
  dw_id="0fafdd43a342436e87e42d7217d26679",
  tb_name="test_update_table_model",
  owner="",
  table_type="HIVE_TABLE",
  configs="{}",
  data_format="Parquet",
  model=modelbody,
  model_id="1208730797675311104",
  id="1231932180633112576"
)
response = client.update_table_model(request)
print(response)
except exceptions.ClientRequestException as e:
  print(e.status_code)
  print(e.request_id)
  print(e.error_code)
  print(e.error_msg)
```

## Go

根据请求参数，更新物理表。

```
package main

import (
    "fmt"
    "time"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/sdktime"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTableModelRequest{}
    var listRelationsbody = []model.RelationVo{
        {
            Name: "relation_name",
        },
    }
    var listMappingsbody = []model.TableMappingVo{
        {
            Name: "mapping_name",
        },
    }
}
```

```

fdNameChSelfDefinedFields:= "苏"
fdNameEnSelfDefinedFields:= "SDSDFD"
notNullSelfDefinedFields:= false
fdValueSelfDefinedFields:= ""
fdNameChSelfDefinedFields1:= "test"
fdNameEnSelfDefinedFields1:= "test"
notNullSelfDefinedFields1:= false
fdValueSelfDefinedFields1:= ""
var listSelfDefinedFieldsAttributes = []model.SelfDefinedFieldVo{
    {
        FdNameCh: &fdNameChSelfDefinedFields,
        FdNameEn: &fdNameEnSelfDefinedFields,
        NotNull: &notNullSelfDefinedFields,
        FdValue: &fdValueSelfDefinedFields,
    },
    {
        FdNameCh: &fdNameChSelfDefinedFields1,
        FdNameEn: &fdNameEnSelfDefinedFields1,
        NotNull: &notNullSelfDefinedFields1,
        FdValue: &fdValueSelfDefinedFields1,
    },
}
var listQualityInfosAttributes = []model.QualityInfoVo{
    {
        DataQualityId: "1231932180754747392",
    },
}
var listSecrecyLevelsAttributes = []model.SecrecyLevelVo{
    {
        Name: "secrecy_level_name",
    },
}
var listTagsAttributes = []model.TagVo{
    {
        Name: "tag1",
    },
}
idAttributes:= "1231932180754747392"
descriptionAttributes:= ""
domainTypeAttributes:= model.GetTableModelAttributeVoDomainTypeEnum().STRING
dataTypeExtendAttributes:= ""
isPartitionKeyAttributes:= false
isForeignKeyAttributes:= false
extendFieldAttributes:= false
notNullAttributes:= false
ordinalAttributes:= int32(1)
dateTimeCreateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "2024-04-22T11:38:40+08:00",
time.UTC)
createTimeAttributes:= sdktime.SdkTime(dateTimeCreateTime)
dateTimeUpdateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "2024-04-22T11:38:40+08:00",
time.UTC)
updateTimeAttributes:= sdktime.SdkTime(dateTimeUpdateTime)
aliasAttributes:= ""
codeAttributes:= ""
var listAttributesbody = []model.TableModelAttributeVo{
    {
        Id: &idAttributes,
        NameEn: "col_a",
        NameCh: "col_a",
        Description: &descriptionAttributes,
        DataType: "STRING",
        DomainType: &domainTypeAttributes,
        DataTypeExtend: &dataTypeExtendAttributes,
        IsPrimaryKey: false,
        IsPartitionKey: &isPartitionKeyAttributes,
        IsForeignKey: &isForeignKeyAttributes,
        ExtendField: &extendFieldAttributes,
        NotNull: &notNullAttributes,
        Ordinal: &ordinalAttributes,
    }
}

```

```

        CreateTime: &createTimeAttributes,
        UpdateTime: &updateTimeAttributes,
        Tags: &listTagsAttributes,
        SecrecyLevels: &listSecrecyLevelsAttributes,
        QualityInfos: &listQualityInfosAttributes,
        Alias: &aliasAttributes,
        SelfDefinedFields: &listSelfDefinedFieldsAttributes,
        Code: &codeAttributes,
    },
}
var listTagsbody = []model.TagVo{
    {
        Name: "tag1",
    },
}
idModel:= "1208730797675311104"
descriptionModel:= ""
isPhysicalModel:= true
frequentModel:= false
topModel:= false
dwTypeModel:= "MRS_HIVE"
dateTimeCreateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "2024-02-18T11:04:39+08:00",
time.UTC)
createTimeModel:= sdktime.SdkTime(dateTimeCreateTime)
dateTimeUpdateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "2024-02-18T11:04:39+08:00",
time.UTC)
updateTimeModel:= sdktime.SdkTime(dateTimeUpdateTime)
createByModel:= "test_uesr"
updateByModel:= "test_uesr"
tableModelPrefixModel:= ""
modelbody := &model.WorkspaceVo{
    Id: &idModel,
    Name: "xingneng2",
    Description: &descriptionModel,
    IsPhysical: &isPhysicalModel,
    Frequent: &frequentModel,
    Top: &topModel,
    DwType: &dwTypeModel,
    CreateTime: &createTimeModel,
    UpdateTime: &updateTimeModel,
    CreateBy: &createByModel,
    UpdateBy: &updateByModel,
    Type: model.GetWorkspaceVoTypeEnum().THIRD_NF,
    TableModelPrefix: &tableModelPrefixModel,
}
hasRelatedLogicTableTableModelUpdateVo:= false
hasRelatedPhysicalTableTableModelUpdateVo:= false
envTypeEnvType:= model.GetEnvTypeEnumEnvTypeEnum().INVALID_TYPE
summaryStatusSummaryStatus:= model.GetSyncStatusEnumSummaryStatusEnum().NO_NEED
dataQualityDataQuality:= model.GetSyncStatusEnumDataQualityEnum().NO_NEED
metaDataLinkMetaDataLink:= model.GetSyncStatusEnumMetaDataTableEnum().NO_NEED
businessAssetBusinessAsset:= model.GetSyncStatusEnumBusinessAssetEnum().NO_NEED
technicalAssetTechnicalAsset:= model.GetSyncStatusEnumTechnicalAssetEnum().NO_NEED
devPhysicalTableDevPhysicalTable:= model.GetSyncStatusEnumDevPhysicalTableEnum().NO_NEED
physicalTablePhysicalTable:= model.GetSyncStatusEnumPhysicalTableEnum().NO_NEED
isPartitionTableModelUpdateVo:= false
dirtyOutSuffixTableModelUpdateVo:= ""
dirtyOutPrefixTableModelUpdateVo:= ""
dirtyOutDatabaseTableModelUpdateVo:= ""
dirtyOutSwitchTableModelUpdateVo:= false
useRecentlyPartitionTableModelUpdateVo:= false
l2IdTableModelUpdateVo:= ""
l1IdTableModelUpdateVo:= "958408897973161984"
l2TableModelUpdateVo:= ""
l1TableModelUpdateVo:= "tesy"
dwNameTableModelUpdateVo:= "MRS_Hive"
dwTypeTableModelUpdateVo:= "MRS_HIVE"
dateTimeUpdateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "2024-04-22T11:38:40+08:00",
time.UTC)

```

```

updateTimeTableModelUpdateVo:= sdktime.SdkTime(dateTimeUpdateTime)
dateTimeCreateTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "2024-04-22T11:38:40+08:00",
time.UTC)
createTimeTableModelUpdateVo:= sdktime.SdkTime(dateTimeCreateTime)
updateByTableModelUpdateVo:= "test_uesr"
createByTableModelUpdateVo:= "test_uesr"
catalogPathTableModelUpdateVo:=
"{\"l1Id\":\"958408897973161984\",\"l2Id\":\"\",\"l3Id\":\"\",\"l4Id\":\"\",\"l5Id\":\"\",\"l6Id\":\"\",\"l7Id\":\"\"}"
bizCatalogIdTableModelUpdateVo:= "958408897973161984"
statusStatus:= model.GetBizStatusEnumStatusEnum().DRAFT
extendInfoTableModelUpdateVo:=
"{\"dirtyOutDatabase\":\"\",\"dirtyOutPrefix\":\"\",\"dirtyOutSuffix\":\"\",\"dirtyOutSwitch\":\"false\"}"
dwldTableModelUpdateVo:= "0fafdd43a342436e87e42d7217d26679"
ownerTableModelUpdateVo:= ""
tableTypeTableModelUpdateVo:= "HIVE_TABLE"
configsTableModelUpdateVo:= "{}"
dataFormatTableModelUpdateVo:= "Parquet"
request.Body = &model.TableModelUpdateVo{
    HasRelatedLogicTable: &hasRelatedLogicTableTableModelUpdateVo,
    HasRelatedPhysicalTable: &hasRelatedPhysicalTableTableModelUpdateVo,
    EnvType: &envTypeEnvType,
    SummaryStatus: &summaryStatusSummaryStatus,
    DataQuality: &dataQualityDataQuality,
    MetaDataLink: &metaDataLinkMetaDataLink,
    BusinessAsset: &businessAssetBusinessAsset,
    TechnicalAsset: &technicalAssetTechnicalAsset,
    DevPhysicalTable: &devPhysicalTableDevPhysicalTable,
    PhysicalTable: &physicalTablePhysicalTable,
    IsPartition: &isPartitionTableModelUpdateVo,
    DirtyOutSuffix: &dirtyOutSuffixTableModelUpdateVo,
    DirtyOutPrefix: &dirtyOutPrefixTableModelUpdateVo,
    DirtyOutDatabase: &dirtyOutDatabaseTableModelUpdateVo,
    DirtyOutSwitch: &dirtyOutSwitchTableModelUpdateVo,
    UseRecentlyPartition: &useRecentlyPartitionTableModelUpdateVo,
    L2Id: &l2IdTableModelUpdateVo,
    L1Id: &l1IdTableModelUpdateVo,
    L2: &l2TableModelUpdateVo,
    L1: &l1TableModelUpdateVo,
    DwName: &dwNameTableModelUpdateVo,
    DwType: &dwTypeTableModelUpdateVo,
    Relations: &listRelationsbody,
    Mappings: &listMappingsbody,
    Attributes: listAttributesbody,
    Tags: &listTagsbody,
    UpdateTime: &updateTimeTableModelUpdateVo,
    CreateTime: &createTimeTableModelUpdateVo,
    UpdateBy: &updateByTableModelUpdateVo,
    CreateBy: &createByTableModelUpdateVo,
    CatalogPath: &catalogPathTableModelUpdateVo,
    BizCatalogId: &bizCatalogIdTableModelUpdateVo,
    Status: &statusStatus,
    Description: "无",
    LogicTbName: "test_update_table_model",
    ExtendInfo: &extendInfoTableModelUpdateVo,
    Dwld: &dwldTableModelUpdateVo,
    TbName: "test_update_table_model",
    Owner: &ownerTableModelUpdateVo,
    TableType: &tableTypeTableModelUpdateVo,
    Configs: &configsTableModelUpdateVo,
    DataFormat: &dataFormatTableModelUpdateVo,
    Model: modelbody,
    ModelId: "1208730797675311104",
    Id: "1231932180633112576",
}
response, err := client.UpdateTableModel(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}

```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是TableModelVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.4 删除表模型

### 功能介绍

在关系建模中删除一个表模型及其属性，包括逻辑实体和物理表。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/table-model

表 8-1427 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-1428 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1429 请求 Body 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	ID列表，ID字符串。
del_types	否	String	删除类型。 枚举值： <ul style="list-style-type: none"> <li>PHYSICAL_TABLE: 是否删除数据库物理表，只针对能物化的表生效（配置此值则表示删除）</li> </ul>

## 响应参数

状态码：200



表 8-1430 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-1431 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

**状态码：400**

表 8-1432 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1433 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1434 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，删除物理表。

```
DELETE https://{endpoint}/v2/{project_id}/design/table-model
{
  "ids" : [ "1231921671028518912" ],
  "del_types" : "PHYSICAL_TABLE"
}
```

## 响应示例

**状态码：200**

Success，返回data数据是删除个数。

```
{
  "value" : 1
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求参数，删除物理表。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteTableModelSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteTableModelRequest request = new DeleteTableModelRequest();
        IdsParam body = new IdsParam();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("1231921671028518912");
        body.withDelTypes(IdsParam.DelTypeEnum.fromValue("PHYSICAL_TABLE"));
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            DeleteTableModelResponse response = client.deleteTableModel(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

根据请求参数，删除物理表。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteTableModelRequest()
    listIdsbody = [
        "1231921671028518912"
    ]
    request.body = IdsParam(
        del_types="PHYSICAL_TABLE",
        ids=listIdsbody
    )
    response = client.delete_table_model(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据请求参数，删除物理表。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteTableModelRequest{}
    var listIdsbody = []string{
        "1231921671028518912",
    }
```

```

}
delTypesIdsParam:= model.GetIdsParamDelTypesEnum().PHYSICAL_TABLE
request.Body = &model.IdsParam{
    DelTypes: &delTypesIdsParam,
    Ids: listIdsbody,
}
response, err := client.DeleteTableModel(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.5 查询关系

### 功能介绍

通过关系名称(支持模糊查询)、创建人、开始时间、结束时间等分页查找关系信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/relation

表 8-1435 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1436 Query 参数

参数	是否必选	参数类型	描述
name	否	String	按名称或编码模糊查询。
create_by	否	String	按创建者查询。
begin_time	否	String	时间过滤左边界，与end_time一起使用，只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
end_time	否	String	时间过滤右边界，与begin_time一起使用只支持时间范围过滤，单边过滤无效。格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-1437 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-1438 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-1439 data

参数	参数类型	描述
value	<b>value</b> object	value, 统一的返回结果的外层数据结构。

表 8-1440 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>RelationVO</b> objects	RelationVO信息。

表 8-1441 RelationVO

参数	参数类型	描述
id	String	编码, ID字符串。
source_table_id	String	源表ID, ID字符串。
target_table_id	String	目标表ID, ID字符串。
name	String	关系名称。

参数	参数类型	描述
source_table_name	String	源表名称。
target_table_name	String	目的表名称。
role	String	角色。
tenant_id	String	租户ID。
source_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
target_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



参数	参数类型	描述
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
mappings	Array of <a href="#">RelationMapping VO</a> objects	表属性信息，只读。

表 8-1442 RelationMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
relation_id	String	关系ID，ID字符串。
source_field_id	String	源字段ID，ID字符串。
target_field_id	String	目标字段ID，ID字符串。
source_field_name	String	源表名称。
target_field_name	String	目的表名称。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-1443 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

### 状态码：401

表 8-1444 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

### 状态码：403

表 8-1445 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求参数，获取关系列表。

```
GET https://{endpoint}/v2/{project_id}/design/relation?offset=1&limit=100
```

## 响应示例

### 状态码：200

Success，返回data数据是RelationVO数组，以及总数。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "id": "1225137038202957824",
        "source_table_id": "1225137036110000128",
        "target_table_id": "1225137037896773632",
        "name": "left_relation_right_test",
        "source_table_name": "test_left_relation",
        "target_table_name": "test_right_test",
        "role": null,
        "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70",
        "source_type": "ONE",
        "target_type": "ONE",
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "create_time": "2024-04-03T17:37:12+08:00",
        "update_time": "2024-04-03T17:37:12+08:00",
        "mappings": [ {
```

```
"id" : "1225137038211346432",
"relation_id" : "1225137038202957824",
"source_field_id" : "1225137036294549504",
"target_field_id" : "1225137037896773633",
"source_field_name" : "id",
"target_field_name" : "id",
"create_by" : "test_uesr",
"update_by" : "test_uesr",
"create_time" : "2024-04-03T17:37:12+08:00",
"update_time" : "2024-04-03T17:37:12+08:00"
  }
}
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListRelationsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
```

```
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListRelationsRequest request = new ListRelationsRequest();
try {
    ListRelationsResponse response = client.listRelations(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListRelationsRequest()
        response = client.list_relations(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListRelationsRequest{}
    response, err := client.ListRelations(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是RelationVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.6 查看关系详情

### 功能介绍

通过ID获取关系详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/relation/{id}

表 8-1446 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-1447 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

### 请求参数

表 8-1448 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-1449 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-1450 data

参数	参数类型	描述
value	<b>RelationVO</b> object	value, 统一的返回结果的外层数据结构。

表 8-1451 RelationVO

参数	参数类型	描述
id	String	编码, ID字符串。
source_table_id	String	源表ID, ID字符串。
target_table_id	String	目标表ID, ID字符串。
name	String	关系名称。
source_table_name	String	源表名称。
target_table_name	String	目的表名称。
role	String	角色。
tenant_id	String	租户ID。

参数	参数类型	描述
source_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
target_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
mappings	Array of <a href="#">RelationMapping VO</a> objects	表属性信息，只读。



表 8-1452 RelationMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
relation_id	String	关系ID，ID字符串。
source_field_id	String	源字段ID，ID字符串。
target_field_id	String	目标字段ID，ID字符串。
source_field_name	String	源表名称。
target_field_name	String	目的表名称。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

状态码：400

表 8-1453 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1454 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1455 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**请求示例**

根据ID查询关系详情，获取关系列表。

GET https://{endpoint}/v2/{project\_id}/design/relation/1225137038202957824

**响应示例**

**状态码：200**

Success，返回data数据是RelationVO详情。

```
{
  "data": {
    "value": {
      "id": "1225137038202957824",
      "source_table_id": "1225137036110000128",
      "target_table_id": "1225137037896773632",
      "name": "left_relation_right_test",
      "source_table_name": "test_left_relation",
      "target_table_name": "test_right_test",
      "role": null,
      "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70",
      "source_type": "ONE",
      "target_type": "ONE",
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "create_time": "2024-04-03T17:37:12+08:00",
      "update_time": "2024-04-03T17:37:12+08:00",
      "mappings": [ {
        "id": "1225137038211346432",
        "relation_id": "1225137038202957824",
        "source_field_id": "1225137036294549504",
        "target_field_id": "1225137037896773633",
        "source_field_name": "id",
        "target_field_name": "id",
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "create_time": "2024-04-03T17:37:12+08:00",
        "update_time": "2024-04-03T17:37:12+08:00"
      } ]
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
```

```
"error_msg" : "The user request is illegal."  
}
```

**状态码：401**

Unauthorized

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "User authentication failed."  
}
```

**状态码：403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowRelationByIdSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowRelationByIdRequest request = new ShowRelationByIdRequest();  
        request.withId("{id}");  
        try {  
            ShowRelationByIdResponse response = client.showRelationById(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowRelationByIdRequest()
        request.id = "{id}"
        response = client.show_relation_by_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```

projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowRelationByIdRequest{}
request.Id = "{id}"
response, err := client.ShowRelationById(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是RelationVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.7 查询模型下所有关系

### 功能介绍

查询模型下所有关系。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/{model\_id}/table-model/relation

表 8-1456 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
model_id	是	String	所属关系建模的模型ID, ID字符串。model_id可从接口 <a href="#">获取模型</a> 中获取。

表 8-1457 Query 参数

参数	是否必选	参数类型	描述
table_ids	否	String	表模型ids, ID字符串。
biz_type	否	String	表类型。 枚举值: <ul style="list-style-type: none"> <li>• TABLE_MODEL: 关系建模: 逻辑实体/物理表</li> <li>• FACT_LOGIC_TABLE: 事实表</li> </ul>
limit	否	Integer	每页查询条数, 即查询Y条数据。默认值50, 取值范围[1,100]。
offset	否	Integer	查询起始坐标, 即跳过X条数据, 仅支持0或limit的整数倍, 不满足则向下取整, 默认值0。

## 请求参数

表 8-1458 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取 ( 响应消息头中X-Subject-Token的值 ) 使用Token认证时必选。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求, 则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

## 响应参数

状态码: 200

表 8-1459 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-1460 data

参数	参数类型	描述
value	<b>value</b> object	value, 统一的返回结果的外层数据结构。

表 8-1461 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>records</b> objects	RelationVO信息。

表 8-1462 records

参数	参数类型	描述
tables	Array of <b>TableModelVO</b> objects	TableModelVO信息。
inheritances	Array of objects	层级信息信息。

参数	参数类型	描述
relations	Array of <b>RelationVO</b> objects	RelationVO信息。

表 8-1463 TableModelVO

参数	参数类型	描述
id	String	编码，ID字符串。
model_id	String	所属关系建模的模型ID，ID字符串。
parent_table_id	String	父表ID，ID字符串。
parent_table_name	String	父表名称，只读。
parent_table_code	String	父表编码，只读。
related_logic_table_id	String	关联逻辑实体的ID，ID字符串。
related_logic_table_name	String	关联逻辑实体的名称。
related_logic_table_model_id	String	关联逻辑实体的模型ID，ID字符串。
related_logic_table_model_name	String	关联逻辑实体的模型名称。
model	<b>WorkspaceVO</b> object	模型信息。
data_format	String	数据格式。
obs_bucket	String	obs桶。
obs_location	String	外表路径
configs	String	其他配置。
table_type	String	表类型，只读。
owner	String	负责人。
tb_name	String	表名。
dw_id	String	数据连接ID。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。



参数	参数类型	描述
schema	String	DWS类型需要。
extend_info	String	扩展信息。
tb_guid	String	表物化后的guid, 只读。
tb_id	String	数据表ID, 只读。
logic_tb_name	String	逻辑实体名。
logic_tb_guid	String	逻辑实体的guid, 只读。
description	String	描述。
status	String	实体的发布状态, 只读, 创建和更新时无需填写。 枚举值: <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
logic_tb_id	String	逻辑实体的ID, ID字符串。
biz_catalog_id	String	归属的业务分类的id, ID字符串。
catalog_path	String	归属的业务分类的路径 { "l1Id": "", "l2Id": "", "l3Id": "" }。
create_by	String	创建人, 只读。
update_by	String	更新人, 只读。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
tags	Array of <a href="#">TagRecordVO</a> objects	表标签, 只读。
approval_info	<a href="#">ApprovalVO</a> object	审批信息, 只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。

参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
attributes	Array of <b>TableModelAttributeVO</b> objects	表属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
relations	Array of <b>RelationVO</b> objects	关系。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，只读，ID字符串。
partition_conf	String	分区表达式。
dlf_task_id	String	DLF作业ID。
use_recently_partition	Boolean	是否使用最新分区。
reversed	Boolean	是否是逆向的。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。

参数	参数类型	描述
dirty_out_suffix	String	异常表后缀。
quality_owner	String	质量责任人。
quality_id	String	质量ID, ID字符串。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值: <ul style="list-style-type: none"> <li>HASH: 对指定的列进行Hash, 通过映射, 把数据分布到指定DN</li> <li>REPLICATION: 表的每一行存在所有数据节点 (DN) 中, 即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
is_partition	Boolean	是否分区表, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>NO_NEED: 未同步</li> <li>CREATE_SUCCESS: 创建成功</li> <li>CREATE_FAILED: 创建失败</li> <li>UPDATE_SUCCESS: 更新成功</li> <li>UPDATE_FAILED: 更新失败</li> <li>SUMMARY_SUCCESS: 整体成功</li> <li>SUMMARY_FAILED: 整体失败</li> <li>RUNNING: 整体运行</li> <li>OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>NO_NEED: 未同步</li> <li>CREATE_SUCCESS: 创建成功</li> <li>CREATE_FAILED: 创建失败</li> <li>UPDATE_SUCCESS: 更新成功</li> <li>UPDATE_FAILED: 更新失败</li> <li>SUMMARY_SUCCESS: 整体成功</li> <li>SUMMARY_FAILED: 整体失败</li> <li>RUNNING: 整体运行</li> <li>OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>• INVALID_TYPE: 无效环境</li> <li>• DEV_TYPE: 开发环境</li> <li>• PROD_TYPE: 生产环境</li> <li>• DEV_PROD_TYPE: 开发生产环境</li> </ul>
alias	String	别名。

参数	参数类型	描述
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
code	String	编码
has_related_physical_table	Boolean	是否存在关联物理表。
has_related_logic_table	Boolean	是否存在关联逻辑实体。

表 8-1464 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>SDI: 贴源数据层</li> <li>DWI: 数据整合层</li> <li>DWR: 数据报告层</li> <li>DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

表 8-1465 TagRecordVO

参数	参数类型	描述
id	String	编码，ID字符串。
tag_id	String	标签ID，ID字符串。
tag_name	String	标签名称。
biz_id	String	实体ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1466 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1467 TableModelAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
obs_location	String	obs路径，子路径。
create_by	String	创建人。
update_by	String	更新人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。

参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键。
extend_field	Boolean	是否继承的属性。
not_null	Boolean	是否不为空。
ordinal	Integer	序号。
table_model_id	String	所属关系建模的模型ID，ID字符串。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
tags	Array of <b>TagVO</b> objects	表标签，只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	String	逻辑属性编码
related_logic_attr_id	String	关联逻辑属性ID，ID字符串。
related_logic_attr_name	String	关联逻辑实体属性中文名称
related_logic_attr_name_en	String	关联逻辑实体属性英文名称

表 8-1468 TagVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	标签名。
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。

表 8-1469 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1470 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>



参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1471 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1472 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1473 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-1474 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>LEFT: 左外连接</li> <li>RIGHT: 右外连接</li> <li>INNER: 内连接</li> <li>FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1475 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-1476 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-1477 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。

参数	参数类型	描述
fd_name_en	String	自定义项英文名。
not_null	Boolean	是否必填。
fd_value	String	属性值。

表 8-1478 RelationVO

参数	参数类型	描述
id	String	编码，ID字符串。
source_table_id	String	源表ID，ID字符串。
target_table_id	String	目标表ID，ID字符串。
name	String	关系名称。
source_table_name	String	源表名称。
target_table_name	String	目的表名称。
role	String	角色。
tenant_id	String	租户ID。
source_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>● ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>● ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>● ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>● ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>



参数	参数类型	描述
target_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
mappings	Array of <a href="#">RelationMapping VO</a> objects	表属性信息，只读。

表 8-1479 RelationMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
relation_id	String	关系ID，ID字符串。
source_field_id	String	源字段ID，ID字符串。
target_field_id	String	目标字段ID，ID字符串。
source_field_name	String	源表名称。
target_field_name	String	目的表名称。
create_by	String	创建人。

参数	参数类型	描述
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

**状态码：400**

表 8-1480 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1481 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1482 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询模型下所有关系，查询前100条。

```
GET https://{endpoint}/v2/{project_id}/design/1208450696123420672/table-model/relation?offset=1&limit=100&no_version_detail=true
```

## 响应示例

状态码：200

Success，返回data数据是RelationVO。

```
{
  "data": {
    "value": {
      "total": 1,
      "records": [ {
        "tables": [ {
          "id": "1204850788338421760",
          "model_id": "1204810091711352832",
          "parent_table_id": null,
          "parent_table_name": null,
          "parent_table_code": null,
          "related_logic_table_id": null,
          "related_logic_table_name": null,
          "related_logic_table_model_id": null,
          "related_logic_table_model_name": null,
          "model": null,
          "data_format": null,
          "obs_bucket": null,
          "obs_location": null,
          "configs": null,
          "table_type": "CLICKHOUSE_TABLE",
          "owner": "",
          "tb_name": "fact_dim_clicknor",
          "dw_id": "f2e4ba7e48df4599a18563f7918fb9ed",
          "db_name": "default",
          "queue_name": null,
          "schema": null,
          "extend_info": "{\"dirtyOutDatabase\":\"\",\"dirtyOutPrefix\":\"\",\"dirtyOutSuffix\":\"\",\"dirtyOutSwitch\":\"false\"}",
          "tb_guid": null,
          "tb_id": null,
          "logic_tb_name": "click正常#( )",
          "logic_tb_guid": null,
          "description": "click正常#( )",
          "status": "DRAFT",
          "logic_tb_id": null,
          "biz_catalog_id": "958408897973161984",
          "catalog_path": null,
          "create_by": "test_uesr",
          "update_by": "test_uesr",
          "create_time": "2024-02-07T18:06:53+08:00",
          "update_time": "2024-02-07T18:06:53+08:00",
          "tags": [ ],
          "approval_info": null,
          "new_biz": null,
          "attributes": [ {
            "id": "1204850788338421761",
            "name_en": "ww",
            "name_ch": "ww",
            "description": null,
            "obs_location": null,
            "create_by": null,
            "update_by": null,
            "data_type": "STRING",
            "domain_type": "STRING",
```

```

    "data_type_extend" : null,
    "is_primary_key" : false,
    "is_partition_key" : false,
    "is_foreign_key" : false,
    "extend_field" : false,
    "not_null" : false,
    "ordinal" : 1,
    "table_model_id" : null,
    "create_time" : "2024-02-07T18:06:53+08:00",
    "update_time" : "2024-02-07T18:06:53+08:00",
    "tags" : [ ],
    "secrecy_levels" : [ ],
    "stand_row_id" : null,
    "stand_row_name" : null,
    "quality_infos" : null,
    "alias" : null,
    "self_defined_fields" : null,
    "code" : null,
    "related_logic_attr_id" : null,
    "related_logic_attr_name" : null,
    "related_logic_attr_name_en" : null
  } ],
  "mappings" : null,
  "relations" : [ ],
  "dw_type" : "CLICKHOUSE",
  "dw_name" : null,
  "l1" : null,
  "l2" : null,
  "l3" : null,
  "l1_id" : null,
  "l2_id" : null,
  "l3_id" : "958408897973161984",
  "partition_conf" : null,
  "dlf_task_id" : null,
  "use_recently_partition" : false,
  "reversed" : null,
  "dirty_out_switch" : false,
  "dirty_out_database" : "",
  "dirty_out_prefix" : "",
  "dirty_out_suffix" : "",
  "quality_owner" : null,
  "quality_id" : null,
  "distribute" : null,
  "distribute_column" : null,
  "is_partition" : false,
  "physical_table" : "NO_NEED",
  "dev_physical_table" : "NO_NEED",
  "technical_asset" : "NO_NEED",
  "business_asset" : "NO_NEED",
  "meta_data_link" : "NO_NEED",
  "data_quality" : "NO_NEED",
  "summary_status" : "NO_NEED",
  "dev_version" : null,
  "prod_version" : null,
  "dev_version_name" : null,
  "prod_version_name" : null,
  "env_type" : "INVALID_TYPE",
  "alias" : null,
  "self_defined_fields" : null,
  "code" : null,
  "has_related_physical_table" : false,
  "has_related_logic_table" : false
  } ],
  "inheritances" : [ ],
  "relations" : [ ]
} ]
}
}
}

```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListTableModelRelationsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListTableModelRelationsRequest request = new ListTableModelRelationsRequest();
        request.withModelId("{model_id}");
    }
}
```

```
try {
    ListTableModelRelationsResponse response = client.listTableModelRelations(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListTableModelRelationsRequest()
        request.model_id = "{model_id}"
        response = client.list_table_model_relations(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListTableModelRelationsRequest{}
request.ModelId = "{model_id}"
response, err := client.ListTableModelRelations(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是RelationVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.8 查看表模型详情

### 功能介绍

通过ID获取模型表详情信息。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/design/table-model/{id}

表 8-1483 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	实体ID，ID字符串。

表 8-1484 Query 参数

参数	是否必选	参数类型	描述
latest	否	Boolean	是否查询最新的。

## 请求参数

表 8-1485 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200



表 8-1486 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-1487 data

参数	参数类型	描述
value	<b>TableModelVO</b> object	value, 统一的返回结果的外层数据结构。

表 8-1488 TableModelVO

参数	参数类型	描述
id	String	编码, ID字符串。
model_id	String	所属关系建模的模型ID, ID字符串。
parent_table_id	String	父表ID, ID字符串。
parent_table_name	String	父表名称, 只读。
parent_table_code	String	父表编码, 只读。
related_logic_table_id	String	关联逻辑实体的ID, ID字符串。
related_logic_table_name	String	关联逻辑实体的名称。
related_logic_table_model_id	String	关联逻辑实体的模型ID, ID字符串。
related_logic_table_model_name	String	关联逻辑实体的模型名称。
model	<b>WorkspaceVO</b> object	模型信息。
data_format	String	数据格式。
obs_bucket	String	obs桶。
obs_location	String	外表路径
configs	String	其他配置。
table_type	String	表类型, 只读。

参数	参数类型	描述
owner	String	负责人。
tb_name	String	表名。
dw_id	String	数据连接ID。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。
extend_info	String	扩展信息。
tb_guid	String	表物化后的guid，只读。
tb_id	String	数据表ID，只读。
logic_tb_name	String	逻辑实体名。
logic_tb_guid	String	逻辑实体的guid，只读。
description	String	描述。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
logic_tb_id	String	逻辑实体的ID，ID字符串。
biz_catalog_id	String	归属的业务分类的id，ID字符串。
catalog_path	String	归属的业务分类的路径 { "l1Id": "", "l2Id": "", "l3Id": "" }。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。

参数	参数类型	描述
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如 1970-01-01T00:00:00Z。
tags	Array of <b>TagRecordVO</b> objects	表标签, 只读。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息, 包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理, 只读。
attributes	Array of <b>TableModelAttributeVO</b> objects	表属性信息。
mappings	Array of <b>TableMappingVO</b> objects	表映射信息。
relations	Array of <b>RelationVO</b> objects	关系。
dw_type	String	数据连接类型, 对应表所在的数仓类型, 取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称, 只读, 创建和更新时无需填写。
l1	String	主题域分组中文名, 只读, 创建和更新时无需填写。
l2	String	主题域中文名, 只读, 创建和更新时无需填写。
l3	String	业务对象中文名, 只读, 创建和更新时无需填写。
l1_id	String	主题域分组ID, 只读, ID字符串。
l2_id	String	主题域ID, 只读, 创建和更新时无需填写。
l3_id	String	业务对象ID, 只读, ID字符串。
partition_conf	String	分区表达式。

参数	参数类型	描述
dlf_task_id	String	DLF作业ID。
use_recently_partition	Boolean	是否使用最新分区。
reversed	Boolean	是否是逆向的。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
quality_owner	String	质量责任人。
quality_id	String	质量ID, ID字符串。
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值: <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash, 通过映射, 把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点 (DN) 中, 即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
is_partition	Boolean	是否分区表, 只读。
physical_table	String	同步状态。 枚举值: <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。

参数	参数类型	描述
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>INVALID_TYPE: 无效环境</li> <li>DEV_TYPE: 开发环境</li> <li>PROD_TYPE: 生产环境</li> <li>DEV_PROD_TYPE: 开发生产环境</li> </ul>
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
code	String	编码
has_related_physical_table	Boolean	是否存在关联物理表。
has_related_logic_table	Boolean	是否存在关联逻辑实体。

表 8-1489 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>SDI: 贴源数据层</li> <li>DWI: 数据整合层</li> <li>DWR: 数据报告层</li> <li>DM: 数据集市层</li> </ul>

参数	参数类型	描述
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

表 8-1490 TagRecordVO

参数	参数类型	描述
id	String	编码，ID字符串。
tag_id	String	标签ID，ID字符串。
tag_name	String	标签名称。
biz_id	String	实体ID，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1491 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1492 TableModelAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
obs_location	String	obs路径，子路径。
create_by	String	创建人。
update_by	String	更新人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。

参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键。
extend_field	Boolean	是否继承的属性。
not_null	Boolean	是否不为空。
ordinal	Integer	序号。
table_model_id	String	所属关系建模的模型ID，ID字符串。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
tags	Array of <b>TagVO</b> objects	表标签，只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	String	逻辑属性编码
related_logic_attr_id	String	关联逻辑属性ID，ID字符串。
related_logic_attr_name	String	关联逻辑实体属性中文名称
related_logic_attr_name_en	String	关联逻辑实体属性英文名称

表 8-1493 TagVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	标签名。
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。

表 8-1494 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1495 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。



参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1496 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1497 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1498 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-1499 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1500 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-1501 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-1502 RelationVO

参数	参数类型	描述
id	String	编码, ID字符串。



参数	参数类型	描述
source_table_id	String	源表ID, ID字符串。
target_table_id	String	目标表ID, ID字符串。
name	String	关系名称。
source_table_name	String	源表名称。
target_table_name	String	目的表名称。
role	String	角色。
tenant_id	String	租户ID。
source_type	String	关系类型, 只读。 枚举值: <ul style="list-style-type: none"> <li>• ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中有且只有一条数据与之对应。</li> <li>• ZERO_OR_ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中最多有一条数据与之对应。</li> <li>• ZERO_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中可能有多条数据与之对应。</li> <li>• ONE_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中至少有一条数据与之对应。</li> </ul>
target_type	String	关系类型, 只读。 枚举值: <ul style="list-style-type: none"> <li>• ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中有且只有一条数据与之对应。</li> <li>• ZERO_OR_ONE: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中最多有一条数据与之对应。</li> <li>• ZERO_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中可能有多条数据与之对应。</li> <li>• ONE_OR_N: 表示每条子(父)逻辑实体数据在父(子)逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
mappings	Array of <b>RelationMappingVO</b> objects	表属性信息，只读。

表 8-1503 RelationMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
relation_id	String	关系ID，ID字符串。
source_field_id	String	源字段ID，ID字符串。
target_field_id	String	目标字段ID，ID字符串。
source_field_name	String	源表名称。
target_field_name	String	目的表名称。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1504 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。

参数	参数类型	描述
not_null	Boolean	是否必填。
fd_value	String	属性值。

状态码：400

表 8-1505 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1506 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1507 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

通过ID获取模型表详情信息。

GET [https://{endpoint}/v2/{project\\_id}/design/table-model/1231932180633112576](https://{endpoint}/v2/{project_id}/design/table-model/1231932180633112576)

## 响应示例

### 状态码：200

Success, 返回data数据是TableModelVO详情。

```
{
  "data": {
    "value": {
      "id": "1231932180633112576",
      "model_id": "1208730797675311104",
      "parent_table_id": null,
      "parent_table_name": null,
      "parent_table_code": null,
      "related_logic_table_id": null,
      "related_logic_table_name": null,
      "related_logic_table_model_id": null,
      "related_logic_table_model_name": null,
      "model": {
        "id": "1208730797675311104",
        "name": "xingneng2",
        "description": "",
        "is_physical": true,
        "frequent": false,
        "top": false,
        "level": null,
        "dw_type": "MRS_HIVE",
        "create_time": "2024-02-18T11:04:39+08:00",
        "update_time": "2024-02-18T11:04:39+08:00",
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "type": "THIRD_NF",
        "biz_catalog_ids": null,
        "databases": null,
        "table_model_prefix": ""
      },
      "data_format": "Parquet",
      "obs_bucket": null,
      "obs_location": null,
      "configs": "{}",
      "table_type": "HIVE_TABLE",
      "owner": "",
      "tb_name": "test_update_table_model",
      "dw_id": "0fafdd43a342436e87e42d7217d26679",
      "db_name": null,
      "queue_name": null,
      "schema": null,
      "extend_info": "{\"dirtyOutDatabase\":\"\",\"dirtyOutPrefix\":\"\",\"dirtyOutSuffix\":\"\",\"dirtyOutSwitch\":"false\"}",
      "tb_guid": null,
      "tb_id": null,
      "logic_tb_name": "test_update_table_model",
      "logic_tb_guid": null,
      "description": "无",
      "status": "DRAFT",
      "logic_tb_id": null,
      "biz_catalog_id": "958408897973161984",
      "catalog_path": "tesy",
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "create_time": "2024-04-22T11:38:40+08:00",
      "update_time": "2024-04-22T11:39:22+08:00",
      "tags": [],
      "approval_info": null,
      "new_biz": null,
      "attributes": [ {
        "id": "1231932180754747392",
        "name_en": "col_a",
        "name_ch": "col_a",

```

```

"description": "",
"obs_location": null,
"create_by": null,
"update_by": null,
"data_type": "STRING",
"domain_type": "STRING",
"data_type_extend": "",
"is_primary_key": false,
"is_partition_key": false,
"is_foreign_key": false,
"extend_field": false,
"not_null": false,
"ordinal": 1,
"table_model_id": null,
"create_time": "2024-04-22T11:38:40+08:00",
"update_time": "2024-04-22T11:39:23+08:00",
"tags": [],
"secrecy_levels": [],
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": [],
"alias": "",
"self_defined_fields": [ {
  "fd_name_ch": "芬",
  "fd_name_en": "SDSDFD",
  "not_null": false,
  "fd_value": ""
}, {
  "fd_name_ch": "test",
  "fd_name_en": "test",
  "not_null": false,
  "fd_value": ""
} ],
"code": "",
"related_logic_attr_id": null,
"related_logic_attr_name": null,
"related_logic_attr_name_en": null
}, {
  "id": "1231932180754747393",
  "name_en": "col_b",
  "name_ch": "col_b",
  "description": "",
  "obs_location": null,
  "create_by": null,
  "update_by": null,
  "data_type": "STRING",
  "domain_type": "STRING",
  "data_type_extend": "",
  "is_primary_key": false,
  "is_partition_key": false,
  "is_foreign_key": false,
  "extend_field": false,
  "not_null": false,
  "ordinal": 2,
  "table_model_id": null,
  "create_time": "2024-04-22T11:38:40+08:00",
  "update_time": "2024-04-22T11:39:23+08:00",
  "tags": [],
  "secrecy_levels": [],
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": [],
  "alias": "",
  "self_defined_fields": [ {
    "fd_name_ch": "芬",
    "fd_name_en": "SDSDFD",
    "not_null": false,
    "fd_value": ""
  }, {

```

```

        "fd_name_ch" : "test",
        "fd_name_en" : "test",
        "not_null" : false,
        "fd_value" : ""
    }],
    "code" : "",
    "related_logic_attr_id" : null,
    "related_logic_attr_name" : null,
    "related_logic_attr_name_en" : null
}],
"mappings" : null,
"relations" : [ ],
"dw_type" : "MRS_HIVE",
"dw_name" : "MRS_Hive",
"l1" : "tesy",
"l2" : null,
"l3" : null,
"l1_id" : "958408897973161984",
"l2_id" : null,
"l3_id" : null,
"partition_conf" : null,
"dlf_task_id" : null,
"use_recently_partition" : false,
"reversed" : null,
"dirty_out_switch" : false,
"dirty_out_database" : "",
"dirty_out_prefix" : "",
"dirty_out_suffix" : "",
"quality_owner" : null,
"quality_id" : null,
"distribute" : null,
"distribute_column" : null,
"is_partition" : false,
"physical_table" : "NO_NEED",
"dev_physical_table" : "NO_NEED",
"technical_asset" : "NO_NEED",
"business_asset" : "NO_NEED",
"meta_data_link" : "NO_NEED",
"data_quality" : "NO_NEED",
"summary_status" : "NO_NEED",
"dev_version" : null,
"prod_version" : null,
"dev_version_name" : null,
"prod_version_name" : null,
"env_type" : "INVALID_TYPE",
"alias" : null,
"self_defined_fields" : null,
"code" : null,
"has_related_physical_table" : false,
"has_related_logic_table" : false
}
}
}

```

**状态码：400**

BadRequest

```

{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}

```

**状态码：401**

Unauthorized

```

{
  "error_code" : "DS.60xx",

```

```
"error_msg" : "User authentication failed."  
}
```

**状态码：403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowTableModelByIdSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowTableModelByIdRequest request = new ShowTableModelByIdRequest();  
        request.withId("{id}");  
        try {  
            ShowTableModelByIdResponse response = client.showTableModelById(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

```
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ShowTableModelByIdRequest()  
        request.id = "{id}"  
        response = client.show_table_model_by_id(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```



```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowTableModelByIdRequest{}
request.Id = "{id}"
response, err := client.ShowTableModelById(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是TableModelIVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.9 获取模型

### 功能介绍

获取当前空间下的全部模型信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/workspaces

表 8-1508 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1509 Query 参数

参数	是否必选	参数类型	描述
workspace_type	否	String	模型工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。
dw_type	否	String	数据连接类型

## 请求参数

表 8-1510 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1511 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data, 统一的返回结果的最外层数据结构。

表 8-1512 data

参数	参数类型	描述
value	<b>value</b> object	value, 统一的返回结果的外层数据结构。

表 8-1513 value

参数	参数类型	描述
total	Integer	总量。
records	Array of <b>WorkspaceVO</b> objects	WorkspaceVO信息。

表 8-1514 WorkspaceVO

参数	参数类型	描述
id	String	编号, ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值: <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>

参数	参数类型	描述
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1515 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1516 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1517 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

获取当前空间下的全部模型信息。

GET https://{endpoint}/v2/{project\_id}/design/workspaces

## 响应示例

状态码：200

Success，返回data数据是WorkspaceVO数组，以及总数。

```
{
  "data": {
    "value": {
      "records": [ {
        "id": "1014218815216717824",
        "name": "test",
        "description": "abc",
        "is_physical": true,
        "frequent": false,
        "top": true,
        "level": "SDI",
        "dw_type": "DWS",
        "create_time": "2022-08-30T17:03:28.822+08:00",
        "update_time": "2022-08-30T17:03:28.822+08:00",
        "create_by": "abc",
        "update_by": "abc",
        "type": "THIRD_NF",
        "biz_catalog_ids": null,
        "databases": null
      } ]
    }
  }
}
```

状态码：400

### BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

### Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

### Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListWorkspacesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListWorkspacesRequest request = new ListWorkspacesRequest();
        try {
            ListWorkspacesResponse response = client.listWorkspaces(request);
            System.out.println(response.toString());
        }
    }
}
```

```
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListWorkspacesRequest()
        response = client.list_workspaces(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListWorkspacesRequest{}
response, err := client.ListWorkspaces(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是WorkspaceVO数组，以及总数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.10 新建模型工作区

### 功能介绍

新建模型工作区。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/workspaces



表 8-1518 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-1519 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1520 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编号，ID字符串。
name	是	String	工作区名字。
description	否	String	描述。
is_physical	否	Boolean	是否为物理表。
frequent	否	Boolean	是否为常用。
top	否	Boolean	分层治理。

参数	是否必选	参数类型	描述
level	否	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	否	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_by	否	String	创建人。
update_by	否	String	更新人。
type	是	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	否	String	关联的业务分层的ID列表 { "l1Ids": [], "l2Ids": [], "l3Ids": [] }。
databases	否	Array of strings	数据库名称数组。
table_model_prefix	否	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

## 响应参数

状态码：200

表 8-1521 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1522 data

参数	参数类型	描述
value	WorkspaceVO object	value, 统一的返回结果的外层数据结构。

表 8-1523 WorkspaceVO

参数	参数类型	描述
id	String	编号, ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值: <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型, 对应表所在的数仓类型, 取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1524 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1525 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1526 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

新建模型工作区。

POST https://{endpoint}/v2/{project\_id}/design/workspaces

```
{
  "name": "test_create_model",
  "table_model_prefix": "",
  "description": "this is a demo.",
  "type": "THIRD_NF",
  "frequent": false,
  "is_physical": "true",
  "dw_type": "DWS"
}
```

## 响应示例

**状态码：200**

Success, 返回data数据是WorkspaceVO。

```
{
  "data": {
    "value": {
      "id": "1231989920109842432",
      "name": "test_create_model",
      "description": "this is a demo.",
      "is_physical": true,
      "frequent": false,
      "top": true,
      "level": null,
      "dw_type": "DWS",
      "create_time": "2024-04-22T15:28:06.22+08:00",
      "update_time": "2024-04-22T15:28:06.22+08:00",
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "type": "THIRD_NF",
      "biz_catalog_ids": null,
      "databases": null,
      "table_model_prefix": ""
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
}
```

```
"error_msg" : "User authentication failed."  
}
```

**状态码: 403**

Forbidden

```
{  
  "error_code" : "DS.60xx",  
  "error_msg" : "The user does not have permission to call this API."  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

新建模型工作区。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class CreateWorkspaceSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateWorkspaceRequest request = new CreateWorkspaceRequest();  
        WorkspaceVO body = new WorkspaceVO();  
        body.withTableModelPrefix("");  
        body.withType(WorkspaceVO.TypeEnum.fromValue("THIRD_NF"));  
        body.withDwType("DWS");  
        body.withFrequent(false);  
        body.withIsPhysical(true);  
        body.withDescription("this is a demo.");  
        body.withName("test_create_model");  
        request.withBody(body);  
        try {  
            CreateWorkspaceResponse response = client.createWorkspace(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

新建模型工作区。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateWorkspaceRequest()
        request.body = WorkspaceVO(
            table_model_prefix="",
            type="THIRD_NF",
            dw_type="DWS",
            frequent=False,
            is_physical=True,
            description="this is a demo.",
            name="test_create_model"
        )
        response = client.create_workspace(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

新建模型工作区。

```
package main

import (
```

```

"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateWorkspaceRequest{}
    tableModelPrefixWorkspaceVo:= ""
    dwTypeWorkspaceVo:= "DWS"
    frequentWorkspaceVo:= false
    isPhysicalWorkspaceVo:= true
    descriptionWorkspaceVo:= "this is a demo."
    request.Body = &model.WorkspaceVo{
        TableModelPrefix: &tableModelPrefixWorkspaceVo,
        Type: model.GetWorkspaceVoTypeEnum().THIRD_NF,
        DwType: &dwTypeWorkspaceVo,
        Frequent: &frequentWorkspaceVo,
        IsPhysical: &isPhysicalWorkspaceVo,
        Description: &descriptionWorkspaceVo,
        Name: "test_create_model",
    }
    response, err := client.CreateWorkspace(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是WorkspaceVO。
400	BadRequest



状态码	描述
401	Unauthorized
403	Forbidden

## 8.22.11 更新模型工作区

### 功能介绍

更新模型工作区。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/workspaces

表 8-1527 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 8-1528 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。

参数	是否必选	参数类型	描述
Content-Type	否	String	默认值: application/json;charset=UTF-8 可选, 有Body体的情况下必选, 没有Body体则无需填写和校验。

表 8-1529 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	编号, ID字符串。
name	是	String	工作区名字。
description	否	String	描述。
is_physical	否	Boolean	是否为物理表。
frequent	否	Boolean	是否为常用。
top	否	Boolean	分层治理。
level	否	String	数据治理分层。 枚举值: <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	否	String	数据连接类型, 对应表所在的数仓类型, 取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_by	否	String	创建人。
update_by	否	String	更新人。
type	是	String	工作区类型枚举。 枚举值: <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	否	String	关联的业务分层的ID列表 { "l1Ids": [], "l2Ids": [], "l3Ids": [] }。

参数	是否必选	参数类型	描述
databases	否	Array of strings	数据库名称数组。
table_model_prefix	否	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

## 响应参数

状态码：200

表 8-1530 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1531 data

参数	参数类型	描述
value	<b>WorkspaceVO</b> object	value，统一的返回结果的外层数据结构。

表 8-1532 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。

参数	参数类型	描述
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1533 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

**状态码：401**

**表 8-1534 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-1535 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

更新模型工作区。

PUT https://{endpoint}/v2/{project\_id}/design/workspaces

```
{
  "id": "1231989920109842432",
  "name": "test_update_model",
  "table_model_prefix": "",
  "description": "this is a demo.",
  "type": "THIRD_NF",
  "frequent": false,
  "is_physical": "true",
  "dw_type": "DWS"
}
```

## 响应示例

**状态码：200**

Success, 返回data数据是WorkspaceVO。

```
{
  "data": {
```

```
"value": {
  "id": "1231989920109842432",
  "name": "test_update_model",
  "description": "this is a demo.",
  "is_physical": true,
  "frequent": false,
  "top": true,
  "level": null,
  "dw_type": "DWS",
  "create_time": "2024-04-22T15:28:06+08:00",
  "update_time": "2024-04-22T15:31:31.913+08:00",
  "create_by": "test_uesr",
  "update_by": "test_uesr",
  "type": "THIRD_NF",
  "biz_catalog_ids": null,
  "databases": null,
  "table_model_prefix": ""
}
```

**状态码：400**

## BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

## Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

## Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新模型工作区。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class UpdateWorkspaceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))
            .build();
        UpdateWorkspaceRequest request = new UpdateWorkspaceRequest();
        WorkspaceVO body = new WorkspaceVO();
        body.withTableModelPrefix("");
        body.withType(WorkspaceVO.TypeEnum.fromValue("THIRD_NF"));
        body.withDwType("DWS");
        body.withFrequent(false);
        body.withIsPhysical(true);
        body.withDescription("this is a demo.");
        body.withName("test_update_model");
        body.withId("1231989920109842432");
        request.withBody(body);
        try {
            UpdateWorkspaceResponse response = client.updateWorkspace(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

更新模型工作区。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
```

```
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateWorkspaceRequest()
    request.body = WorkspaceVO(
        table_model_prefix="",
        type="THIRD_NF",
        dw_type="DWS",
        frequent=False,
        is_physical=True,
        description="this is a demo.",
        name="test_update_model",
        id="1231989920109842432"
    )
    response = client.update_workspace(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

更新模型工作区。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateWorkspaceRequest{
        tableModelPrefixWorkspaceVo: ""
        dwTypeWorkspaceVo: "DWS"
    }
```



```
frequentWorkspaceVo:= false
isPhysicalWorkspaceVo:= true
descriptionWorkspaceVo:= "this is a demo."
idWorkspaceVo:= "1231989920109842432"
request.Body = &model.WorkspaceVo{
    TableModelPrefix: &tableModelPrefixWorkspaceVo,
    Type: model.GetWorkspaceVoTypeEnum().THIRD_NF,
    DwType: &dwTypeWorkspaceVo,
    Frequent: &frequentWorkspaceVo,
    IsPhysical: &isPhysicalWorkspaceVo,
    Description: &descriptionWorkspaceVo,
    Name: "test_update_model",
    Id: &idWorkspaceVo,
}
response, err := client.UpdateWorkspace(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是WorkspaceVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.12 删除模型工作区

### 功能介绍

删除模型工作区。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/workspaces

表 8-1536 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1537 Query 参数

参数	是否必选	参数类型	描述
ids	是	Array of strings	实体ID数组，ID字符串。

## 请求参数

表 8-1538 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1539 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	删除的最终返回结果，返回成功删除的对象个数。

表 8-1540 data

参数	参数类型	描述
value	Integer	成功删除的对象个数

**状态码：400**

表 8-1541 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1542 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1543 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据ID删除模型工作区。

```
DELETE https://{endpoint}/v2/{project_id}/design/workspaces?ids=1231989920109842432
```

## 响应示例

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteWorkspacesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
    }
}
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteWorkspacesRequest request = new DeleteWorkspacesRequest();
try {
    DeleteWorkspacesResponse response = client.deleteWorkspaces(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteWorkspacesRequest()
        response = client.delete_workspaces(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteWorkspacesRequest{}
    response, err := client.DeleteWorkspaces(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是删除个数。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.13 查询模型详情

### 功能介绍

查询物理模型或逻辑模型的工作区空间详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/workspaces/{model\_id}

表 8-1544 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
model_id	是	String	所属关系建模的模型ID，ID字符串。model_id可从接口 <a href="#">获取模型</a> 中获取。

### 请求参数

表 8-1545 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1546 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1547 data

参数	参数类型	描述
value	<b>WorkspaceVO</b> object	value，统一的返回结果的外层数据结构。

表 8-1548 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。



参数	参数类型	描述
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

**状态码：400**

**表 8-1549 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-1550 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

**状态码：403**

表 8-1551 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询物理模型或逻辑模型的工作区空间详情。

GET https://{endpoint}/v2/{project\_id}/design/workspaces/1208730797675311104

## 响应示例

**状态码：200**

Success, 返回data数据是WorkspaceVO详情。

```
{
  "data": {
    "value": {
      "id": "1208730797675311104",
      "name": "xingneng2",
      "description": "",
      "is_physical": true,
      "frequent": false,
      "top": false,
      "level": null,
      "dw_type": "MRS_HIVE",
      "create_time": "2024-02-18T11:04:39+08:00",
      "update_time": "2024-02-18T11:04:39+08:00",
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "type": "THIRD_NF",
      "biz_catalog_ids": null,
      "databases": [ "bigdatatest" ],
      "table_model_prefix": ""
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

**SDK 代码示例**

SDK代码示例如下。

**Java**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowWorkspaceDetailByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowWorkspaceDetailByIdRequest request = new ShowWorkspaceDetailByIdRequest();
        request.withModelId("{model_id}");
        try {
            ShowWorkspaceDetailByIdResponse response = client.showWorkspaceDetailById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

if `__name__ == "__main__"`:

# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables `CLOUD_SDK_AK` and `CLOUD_SDK_SK` in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowWorkspaceDetailByIdRequest()
    request.model_id = "{model_id}"
    response = client.show_workspace_detail_by_id(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
```

```

WithAk(ak).
WithSk(sk).
WithProjectId(projectId).
Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowWorkspaceDetailByIdRequest{}
request.ModelId = "{model_id}"
response, err := client.ShowWorkspaceDetailById(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是WorkspaceVO详情。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.14 查询目的表和字段(待下线)

### 功能介绍

查询目的表和字段(待下线)。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/{model\_id}/table-model/relation/fields

表 8-1552 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
model_id	是	String	所属关系建模的模型ID，ID字符串。model_id可从接口 <a href="#">获取模型</a> 中获取。

表 8-1553 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。

## 请求参数

表 8-1554 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1555 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：400

表 8-1556 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1557 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1558 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

无

## 响应示例

**状态码：200**

Success

```
{
  "data" : {
    "value" : [ "abc" ]
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchFieldsForRelationSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```



```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
SearchFieldsForRelationRequest request = new SearchFieldsForRelationRequest();
request.setModelId("{model_id}");
try {
    SearchFieldsForRelationResponse response = client.searchFieldsForRelation(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchFieldsForRelationRequest()
        request.model_id = "{model_id}"
        response = client.search_fields_for_relation(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchFieldsForRelationRequest{}
    request.ModelId = "{model_id}"
    response, err := client.SearchFieldsForRelation(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.15 导出模型中表的 DDL 语句

### 功能介绍

根据模型ID导出指定表的DDL语句。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/workspaces/{model\_id}/export

表 8-1559 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
model_id	是	String	所属关系建模的模型ID，ID字符串。model_id可从接口 <a href="#">获取模型</a> 中获取。

表 8-1560 Query 参数

参数	是否必选	参数类型	描述
tb_names	否	Array of strings	待导出的表名。
with_db	否	Boolean	导出的DDL包不包括数据库名。

### 请求参数

表 8-1561 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1562 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	返回的数据信息。

表 8-1563 data

参数	参数类型	描述
value	String	接口导出的表的DDL语句。

状态码：400

表 8-1564 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1565 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1566 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

导出模型ID为1217123720355803136中表名为class、trl\_aSCb01c的表的DDL，DDL中包含库名。

```
GET https://{endpoint}/v2/{project_id}/design/workspaces/1217123720355803136/export?with_db=true&tb_names=class&tb_names=trl_aSCb01c
```

## 响应示例

状态码：200

Success，返回data数据是DDL文本信息。

```
{
  "data": {
    "value": "CREATE TABLE IF NOT EXISTS ${database}.class (\n      class_id BIGINT NOT NULL comment '1111',\n      class_name VARCHAR(1024) NOT NULL comment '2222',\n      teacher VARCHAR(255) comment '3333'\n    )\n    PRIMARY KEY(class_id)\n  )\n  ALTER TABLE ${database}.pd_test COMMENT '无'\n  CREATE TABLE IF NOT EXISTS ${database}.logic_{$A_2} (\n      id VARCHAR(255) \n    )\n  ALTER TABLE ${database}.logic_{$A_2} COMMENT 'aaa';"
```

状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ExportDesignModelTableDdlSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ExportDesignModelTableDdlRequest request = new ExportDesignModelTableDdlRequest();
        request.withModelId("{model_id}");
        try {
            ExportDesignModelTableDdlResponse response = client.exportDesignModelTableDdl(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ExportDesignModelTableDdlRequest()  
        request.model_id = "{model_id}"  
        response = client.export_design_model_table_ddl(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ExportDesignModelTableDdlRequest{}
request.ModelId = "{model_id}"
response, err := client.ExportDesignModelTableDdl(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是DDL文本信息。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.16 转换逻辑模型为物理模型

### 功能介绍

转换逻辑模型为物理模型，转换成功则显示转换后的目标模型信息。

异常：目标模型信息的“id”等属性为null时，则需要调用《获取操作结果》接口查看具体报错信息：GET https://{endpoint}/v1/{project\_id}/design/operation-results?operation\_type=TRANSFORM\_LOGIC\_MODEL&operation\_id={model\_id}，其中{model\_id}为本接口的路径参数。

约束与限制：

- 1、target\_model\_id与target\_model\_name不对应时，target\_model\_id优先级高于target\_model\_name。
- 2、target\_model\_id与dw\_type不对应时，会找不到模型，提示：模型不存在。

### 调用方法

请参见[如何调用API](#)。



## URI

PUT /v1/{project\_id}/design/workspaces/{model\_id}/transform

表 8-1567 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
model_id	是	String	所属关系建模的模型ID，ID字符串。model_id可从接口 <a href="#">获取模型</a> 中获取。

## 请求参数

表 8-1568 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1569 请求 Body 参数

参数	是否必选	参数类型	描述
target_model_id	否	String	关系建模的目标模型ID，ID字符串。 注意：当使用已经存在的target_model_name时，该值为必填项，且要与已存在的模型ID保持一致。
target_model_name	是	String	工作区名字。
description	否	String	描述。
update_exist_tables	否	Boolean	是否更新已有表。
ids	否	Array of strings	需要物化的逻辑实体的ID列表，ID字符串。
dw_type	是	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
connection_id	否	String	转化后物理表所属的数据连接ID。
database	否	String	转化后物理表所属的数据库。
queue	否	String	转化后物理表所属的队列（仅DLI）。
schema	否	String	转化后物理表所属的schema（仅DWS和PostgreSQL）。

## 响应参数

状态码：200

表 8-1570 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1571 data

参数	参数类型	描述
value	<b>WorkspaceVO</b> object	value, 统一的返回结果的外层数据结构。

表 8-1572 WorkspaceVO

参数	参数类型	描述
id	String	编号, ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值: <ul style="list-style-type: none"> <li>• SDI: 贴源数据层</li> <li>• DWI: 数据整合层</li> <li>• DWR: 数据报告层</li> <li>• DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型, 对应表所在的数仓类型, 取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

状态码：400

表 8-1573 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1574 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1575 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

将model\_id为1184439352260976640的逻辑模型转化为物理模型。

PUT https://{endpoint}/v1/{project\_id}/design/workspaces/1184439352260976640/transform

```
{
  "target_model_id": "1234524753512017920",
  "target_model_name": "dws0429",
  "update_exist_tables": false,
  "description": "",
  "dw_type": "DWS",
  "connection_id": "92558783022c4ac88965e045b7c7078c",
  "database": "gaussdb",
  "schema": "public",
  "ids": [ "1238437092523188266" ]
}
```

## 响应示例

**状态码：200**

Success, 返回data数据是目标物理WorkspaceVO信息, 最终的转换结果需要从获取批量操作结果的接口中获取。

```
{
  "data": {
    "value": {
      "id": "1234524753512017920",
      "name": "dws0429",
      "description": "",
      "is_physical": true,
      "frequent": false,
      "top": true,
      "level": null,
      "dw_type": "DWS",
      "create_time": "2024-04-29T15:20:38+08:00",
      "update_time": "2024-04-29T15:20:38+08:00",
      "create_by": "test_uesr",
      "update_by": "test_uesr",
      "type": "THIRD_NF",
      "biz_catalog_ids": null,
      "databases": null,
      "table_model_prefix": ""
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

将model\_id为1184439352260976640的逻辑模型转化为物理模型。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateDesignTableModelsFromLogicSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchCreateDesignTableModelsFromLogicRequest request = new
        BatchCreateDesignTableModelsFromLogicRequest();
        request.withModelId("{model_id}");
        WorkspaceTransformVO body = new WorkspaceTransformVO();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("1238437092523188266");
        body.withSchema("public");
        body.withDatabase("gaussdb");
        body.withConnectionId("92558783022c4ac88965e045b7c7078c");
        body.withDwType("DWS");
    }
}
```

```
body.withIds(listbodyIds);
body.withUpdateExistTables(false);
body.withDescription("");
body.withTargetModelName("dws0429");
body.withTargetModelId("1234524753512017920");
request.withBody(body);
try {
    BatchCreateDesignTableModelsFromLogicResponse response =
client.batchCreateDesignTableModelsFromLogic(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

将model\_id为1184439352260976640的逻辑模型转化为物理模型。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateDesignTableModelsFromLogicRequest()
        request.model_id = "{model_id}"
        listIdsbody = [
            "1238437092523188266"
        ]
        request.body = WorkspaceTransformVO(
            schema="public",
            database="gaussdb",
            connection_id="92558783022c4ac88965e045b7c7078c",
            dw_type="DWS",
            ids=listIdsbody,
            update_exist_tables=False,
            description="",
            target_model_name="dws0429",
            target_model_id="1234524753512017920"
        )
```

```
)
response = client.batch_create_design_table_models_from_logic(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

将model\_id为1184439352260976640的逻辑模型转化为物理模型。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateDesignTableModelsFromLogicRequest{}
    request.ModelId = "{model_id}"
    var listIdsbody = []string{
        "1238437092523188266",
    }
    schemaWorkspaceTransformVo:= "public"
    databaseWorkspaceTransformVo:= "gaussdb"
    connectionIdWorkspaceTransformVo:= "92558783022c4ac88965e045b7c7078c"
    updateExistTablesWorkspaceTransformVo:= false
    descriptionWorkspaceTransformVo:= ""
    targetModelNameWorkspaceTransformVo:= "dws0429"
    targetModelIdWorkspaceTransformVo:= "1234524753512017920"
    request.Body = &model.WorkspaceTransformVo{
        Schema: &schemaWorkspaceTransformVo,
        Database: &databaseWorkspaceTransformVo,
        ConnectionId: &connectionIdWorkspaceTransformVo,
        DwType: "DWS",
        Ids: &listIdsbody,
        UpdateExistTables: &updateExistTablesWorkspaceTransformVo,
        Description: &descriptionWorkspaceTransformVo,
        TargetModelName: &targetModelNameWorkspaceTransformVo,
        TargetModelId: &targetModelIdWorkspaceTransformVo,
    }
    response, err := client.BatchCreateDesignTableModelsFromLogic(request)
```



```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是目标物理WorkspaceVO信息，最终的转换结果需要从获取批量操作结果的接口中获取。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.22.17 获取操作结果

### 功能介绍

获取批量操作的结果，如逻辑模型转物理模型和逆向数据库操作。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/design/operation-results

表 8-1576 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1577 Query 参数

参数	是否必选	参数类型	描述
operation_type	是	String	批量操作类型。 枚举值： <ul style="list-style-type: none"> <li>ER_REVERSE_DB: 关系建模逆向数据库</li> <li>TRANSFORM_LOGIC_MODEL: 逻辑模型转物理模型</li> </ul>
limit	否	Integer	每页查询条数，即查询Y条数据。默认值50，取值范围[1,100]。
offset	否	Integer	查询起始坐标，即跳过X条数据，仅支持0或limit的整数倍，不满足则向下取整，默认值0。
operation_id	否	String	批量操作id，在逻辑模型转物理表时，填写的是逻辑模型的model_id，在逆向数据库时，填写的是目标模型的model_id。model_id可从接口 <a href="#">获取模型</a> 中获取。

## 请求参数

表 8-1578 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1579 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1580 data

参数	参数类型	描述
value	<b>BatchOperationVO</b> object	value，统一的返回结果的外层数据结构。

表 8-1581 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID，ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID，ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID，ID字符串。
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 运行中</li> <li>• SUCCESS: 操作成功</li> <li>• FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。

参数	参数类型	描述
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。
groups	Array of <b>BatchOperationVO</b> objects	分组信息。

**状态码：400**

表 8-1582 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1583 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1584 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

在model\_id为1184439352260976640的逻辑模型进行逻辑模型转物理模型操作后，获取转换结果。

```
GET https://{endpoint}/v1/{project_id}/design/operation-results?
operation_type=TRANSFORM_LOGIC_MODEL&operation_id=1184439352260976640
```

## 响应示例

**状态码：200**

Success, 返回data数据是 BatchOperationVO

```
{
  "data": {
    "value": {
      "id": null,
      "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70",
      "group_id": "1243524510200836096",
      "biz_name": null,
      "biz_id": null,
      "operation_status": "SUCCESS",
      "operation_type": "TRANSFORM_LOGIC_MODEL",
      "biz_info": null,
      "create_by": "test_uesr",
      "remark": null,
      "total": 1,
      "success": 1,
      "failed": 0,
      "rate": "1.00",
      "logs": null,
      "groups": [ {
        "id": "1243524510620266496",
        "tenant_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-eeb055e69c624311b6b9cfee89a4ec70",
        "group_id": "1243524510200836096",
        "biz_name": "ljst202",
        "biz_id": null,
        "operation_status": "SUCCESS",
        "operation_type": "TRANSFORM_LOGIC_MODEL",
        "biz_info": null,
        "create_by": "test_uesr",
        "remark": "SKIP | 用户跳过",
        "total": 0,
        "success": 0,
        "failed": 0,
        "rate": null,
        "logs": null,
        "groups": null
      } ]
    }
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowDesignOperationResultSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDesignOperationResultRequest request = new ShowDesignOperationResultRequest();
        try {
            ShowDesignOperationResultResponse response = client.showDesignOperationResult(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ShowDesignOperationResultRequest()  
        response = client.show_design_operation_result(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dataartsstudio.NewDataArtsStudioClient(  

```

```
dataartsstudio.DataArtsStudioClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build()

request := &model.ShowDesignOperationResultRequest{}
response, err := client.ShowDesignOperationResult(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是 BatchOperationVO
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.23 导入导出接口

### 8.23.1 导入模型，关系建模，维度建模，码表，业务指标以及流程架构

#### 功能介绍

导入模型，关系建模，维度建模，码表，业务指标以及流程架构。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v2/{project\_id}/design/models/action



表 8-1585 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1586 Query 参数

参数	是否必选	参数类型	描述
action-id	是	String	<p>需要执行的动作，根据导入的对象不同而选择不同的导入动作。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>import_relation: 导入关系模型：逻辑实体/物理表</li> <li>import_dimension: 导入维度表、事实表</li> <li>import_codetable: 导入码表</li> <li>import_datastandard: 导入数据标准</li> <li>import_bizmetric: 导入业务指标</li> <li>import_bizcatalog: 导入流程架构</li> <li>import_atomic: 导入原子指标</li> <li>import_derivative: 导入衍生指标</li> <li>import_compound: 导入复合指标</li> <li>import_aggregation: 导入汇总表</li> </ul>
model_id	否	String	关系建模的模型ID，在导入模型（import_relation）时必填。
directory_id	否	String	导入的目录id，在导入码表（import_codetable）和数据标准（import_datastandard）时生效，选填。
skip_exist	否	Boolean	是否需要覆盖更新已有的实体。

## 请求参数

表 8-1587 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。
X-Language	否	String	默认值：en-us 可选，导入导出接口必填，可选值有：zh-cn、en-us，分别表示中文、英文。

表 8-1588 FormData 参数

参数	是否必选	参数类型	描述
file	是	File	导入用的excel文件，大小必须小于4MB，且行数小于3000行。

## 响应参数

状态码：200

表 8-1589 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回的数据信息。

表 8-1590 data

参数	参数类型	描述
uuid	String	导入接口返回的唯一标识。

状态码：400

表 8-1591 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：401

表 8-1592 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：403

表 8-1593 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

上传模型excel，通过form表单提交，file为具体文件。

```
POST https://{endpoint}/v2/{project_id}/design/models/action?action-id=import_relation&skip_exist=false&model_id=1208730797675311104
```

## 响应示例

状态码：200

Success，返回data数据是导入任务的uuid，可根据该uuid去查询导入结果。

```
{  
  "data" : {  
    "uuid" : "82f70d35-f61a-46dc-a245-0b86905e82d1"  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ImportModelsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ImportModelsRequest request = new ImportModelsRequest();
        try {
            ImportModelsResponse response = client.importModels(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ImportModelsRequest()
    response = client.import_models(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ImportModelsRequest{}
    response, err := client.ImportModels(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是导入任务的uuid，可根据该uuid去查询导入结果。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.23.2 导入主题

### 功能介绍

用于导入主题。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/catalogs/action

表 8-1594 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1595 Query 参数

参数	是否必选	参数类型	描述
action-id	是	LIST<STRING>	需要执行的动作。 枚举值： <ul style="list-style-type: none"><li>start-import: 开始导入</li></ul>
skip_exist	否	Boolean	是否需要覆盖更新已有的主题。

## 请求参数

表 8-1596 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1597 FormData 参数

参数	是否必选	参数类型	描述
file	是	File	导入主题用的excel文件，大小必须小于4MB，且行数小于3000行。

## 响应参数

状态码：200

表 8-1598 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	返回的数据信息。

表 8-1599 data

参数	参数类型	描述
uuid	String	导入接口返回的唯一标识。

状态码：400

表 8-1600 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：401

表 8-1601 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：403

表 8-1602 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

上传主题excel，通过form表单提交，file为具体excel文件。

POST [https://{endpoint}/v2/{project\\_id}/design/catalogs/action?action-id=start-import&skip\\_exist=false](https://{endpoint}/v2/{project_id}/design/catalogs/action?action-id=start-import&skip_exist=false)

## 响应示例

状态码：200

Success，返回data数据是导入任务的uuid，可根据该uuid去查询导入结果。

```
{
  "data": {
    "uuid": "82f70d35-f61a-46dc-a245-0b86905e82d1"
  }
}
```



## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ImportCatalogsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ImportCatalogsRequest request = new ImportCatalogsRequest();
        try {
            ImportCatalogsResponse response = client.importCatalogs(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ImportCatalogsRequest()
    response = client.import_catalogs(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ImportCatalogsRequest{}
    response, err := client.ImportCatalogs(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是导入任务的uuid，可根据该uuid去查询导入结果。
400	BadRequest
401	Unauthorized
403	Forbidden

### 8.23.3 查询导入结果

#### 功能介绍

查询导入excel的处理结果（其中参数uuid获取为：`/design/models/action`或`/design/catalogs/action`接口返回结果）。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET `/v2/{project_id}/design/catalogs/process-import`

表 8-1603 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1604 Query 参数

参数	是否必选	参数类型	描述
uuid	是	String	需要查询的某次导入的处理结果。

## 请求参数

表 8-1605 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1606 响应 Body 参数

参数	参数类型	描述
data	<a href="#">DsProcessImportResultVO</a> object	data，统一的返回结果的最外层数据结构。

表 8-1607 DsProcessImportResultVO

参数	参数类型	描述
status	String	标识本次导入的唯一值，用于查询导入结果。 枚举值： <ul style="list-style-type: none"> <li>importing: 导入中</li> <li>fail: 导入失败</li> <li>success: 导入成功</li> </ul>

参数	参数类型	描述
group	BatchOperationVO object	分组信息。
rate	String	当前进度。

表 8-1608 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID, ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID, ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID, ID字符串。
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值: <ul style="list-style-type: none"> <li>• RUNNING: 运行中</li> <li>• SUCCESS: 操作成功</li> <li>• FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。
groups	Array of BatchOperationVO objects	分组信息。

状态码: 400

表 8-1609 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：401

表 8-1610 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：403

表 8-1611 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

根据uuid查询导入excel的处理结果。

```
GET https://{endpoint}/v2/{project_id}/design/catalogs/process-import?uuid=5a8b96cc-6ee1-429b-b6a9-b26b5f669218
```

## 响应示例

状态码：200

Success, 返回data数据是DsProcessImportResultVO。

```
{
  "data": {
    "status": "success",
    "group": {
      "id": null,
      "tenant_id": "xxx",
      "group_id": "1014919548664713216",
      "biz_name": null,
      "biz_id": null,
      "operation_status": "SUCCESS",
      "operation_type": "IMPORT_CODETABLE",
      "biz_info": null,
      "create_by": "abc",
      "remark": null,
      "total": 2,
      "success": 2,
      "failed": 0,
      "rate": "1.00",
      "logs": null,
    }
  }
}
```

```
"groups" : [ {
  "id" : "1014919548861845504",
  "tenant_id" : "xxx",
  "group_id" : "1014919548664713216",
  "biz_name" : "test",
  "biz_id" : null,
  "operation_status" : "SUCCESS",
  "operation_type" : "IMPORT_CODETABLE",
  "biz_info" : "{\"op_name\":\"test\",\"op_number\":0}",
  "create_by" : "abc",
  "remark" : "SKIP",
  "total" : 0,
  "success" : 0,
  "failed" : 0,
  "rate" : null,
  "logs" : null,
  "groups" : null
}]
},
"rate" : "1.00"
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ImportResultSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ImportResultRequest request = new ImportResultRequest();
        try {
            ImportResultResponse response = client.importResult(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ImportResultRequest()
        response = client.import_result(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```



```

projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ImportResultRequest{}
response, err := client.ImportResult(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success，返回data数据是DsProcessImportResultVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.23.4 导出业务数据

### 功能介绍

根据请求参数，导出业务数据，可以导出：码表、数据标准、原子指标、衍生指标、复合指标、汇总表、业务指标、主题、流程、逻辑模型、物理模型、维度、事实表。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/design/export-model

表 8-1612 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-1613 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1614 请求 Body 参数

参数	是否必选	参数类型	描述
ids	否	Array of strings	导出对象ID的列表，如：某几个逻辑模型的ID，ID字符串。
asyn	否	Boolean	是否是异步操作导出，true:异步，false:同步。

参数	是否必选	参数类型	描述
type	否	String	导出的业务类型：ER(关系建模)，Directory_CodeTable(码表目录)，Directory_Standard(标准目录)，DIM(维度建模)，codeTable(码表);dataStandard数据标准;directory_id导出指定目录下的码表/数据标准;model_id，biz_catalog_id导出指定模型，目录下的业务表，import_bizcatalog导出流程架构，import_bizmetric导出业务指标。
directory_id	否	String	所属目录ID，ID字符串。
biz_catalog_id	否	String	所属业务分层的ID，ID字符串。
biz_catalog_id_list	否	Array of strings	所属的业务分层的ID列表。
model_id	否	String	所属关系建模的模型ID，导出关系模型需要此参数，ID字符串。

## 响应参数

状态码：200

表 8-1615 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回的数据信息。

表 8-1616 data

参数	参数类型	描述
uuid	String	导入接口返回的唯一标识。

状态码：400

表 8-1617 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1618 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1619 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

导出关系建模中，模型ID为1213499006739791872，表ID为1226583503525818368、1217121407599800322的业务数据。

POST https://{endpoint}/v2/{project\_id}/design/export-model

```
{
  "asyn": "true",
  "type": "ER",
  "model_id": "1213499006739791872",
  "ids": [ "1226583503525818368", "1217121407599800322" ]
}
```

## 响应示例

状态码：200

Success

```
{
  "data": {
    "uuid": "fd653968-864b-45f6-9842-2b125d2dbfb7"
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

导出关系建模中，模型ID为1213499006739791872，表ID为1226583503525818368、1217121407599800322的业务数据。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ExportDesignModelsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ExportDesignModelsRequest request = new ExportDesignModelsRequest();
ExportParam body = new ExportParam();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1226583503525818368");
listbodyIds.add("1217121407599800322");
body.withModelId("1213499006739791872");
body.withType("ER");
body.withAsyn(true);
body.withIds(listbodyIds);
request.withBody(body);
try {
    ExportDesignModelsResponse response = client.exportDesignModels(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

导出关系建模中，模型ID为1213499006739791872，表ID为1226583503525818368、1217121407599800322的业务数据。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ExportDesignModelsRequest()
```

```
listIdsbody = [
    "1226583503525818368",
    "1217121407599800322"
]
request.body = ExportParam(
    model_id="1213499006739791872",
    type="ER",
    asyn=True,
    ids=listIdsbody
)
response = client.export_design_models(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

导出关系建模中，模型ID为1213499006739791872，表ID为1226583503525818368、1217121407599800322的业务数据。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExportDesignModelsRequest{}
    var listIdsbody = []string{
        "1226583503525818368",
        "1217121407599800322",
    }
    modelIdExportParam:= "1213499006739791872"
    typeExportParam:= "ER"
    asynExportParam:= true
    request.Body = &model.ExportParam{
        ModelId: &modelIdExportParam,
        Type: &typeExportParam,
        Asyn: &asynExportParam,
        Ids: &listIdsbody,
    }
}
```

```
response, err := client.ExportDesignModels(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.23.5 获取 excel 导出结果

### 功能介绍

根据请求导出业务数据（/export-model）时返回的uuid，获取excel导出结果。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/design/export-result

表 8-1620 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1621 Query 参数

参数	是否必选	参数类型	描述
uuid	是	String	请求导出接口时返回的uuid。



## 请求参数

表 8-1622 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1623 响应 Body 参数

参数	参数类型	描述
data	<a href="#">data</a> object	导出结果对象。

表 8-1624 data

参数	参数类型	描述
status	String	标识本次导出的唯一值，用于查询导入结果。importing(导出中)、fail(导出失败)、success(导出成功)。 枚举值： <ul style="list-style-type: none"> <li>importing: 导出中</li> <li>fail: 导出失败</li> <li>success: 导出成功</li> </ul>

参数	参数类型	描述
group	<b>BatchOperationVO</b> object	分组信息，导出时可忽略。
rate	String	当前进度。

表 8-1625 BatchOperationVO

参数	参数类型	描述
id	String	批量审批ID，ID字符串。
tenant_id	String	项目ID。
group_id	String	组ID，ID字符串。
biz_name	String	业务名。
biz_id	String	业务ID，ID字符串。
operation_status	String	操作结果类型枚举。RUNNING(运行中)、SUCCESS(操作成功)、FAILED(操作失败)。 枚举值： <ul style="list-style-type: none"> <li>• RUNNING: 运行中</li> <li>• SUCCESS: 操作成功</li> <li>• FAILED: 操作失败</li> </ul>
operation_type	String	类型。
biz_info	String	业务详情。
create_by	String	创建人。
remark	String	remark信息。
total	Integer	总数。
success	Integer	操作成功个数。
failed	Integer	操作失败个数。
rate	String	当前进度。
logs	String	日志。
groups	Array of <b>BatchOperationVO</b> objects	分组信息。

状态码：400

表 8-1626 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：401

表 8-1627 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

状态码：403

表 8-1628 响应 Body 参数

参数	参数类型	描述
data	Object	返回的数据信息。

## 请求示例

根据/export-model接口返回的uuid(fd653968-864b-45f6-9842-2b125d2dbfb7)查询导出结果，查询中返回的是导出状态，导出结束后返回的是二进制文件。

```
GET https://{endpoint}/v2/{project_id}/design/export-result?uuid=fd653968-864b-45f6-9842-2b125d2dbfb7
```

## 响应示例

状态码：200

Success，返回data数据是DsProcessImportResultVO。

```
{
  "data": {
    "status": "importing",
    "group": null,
    "rate": "0"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ExportDesignResultSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ExportDesignResultRequest request = new ExportDesignResultRequest();
        try {
            ExportDesignResultResponse response = client.exportDesignResult(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ExportDesignResultRequest()
    response = client.export_design_result(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExportDesignResultRequest{}
    response, err := client.ExportDesignResult(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是DsProcessImportResultVO。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.24 自定义项接口

### 8.24.1 查询自定义项

#### 功能介绍

查询自定义项（包括表自定义项、属性自定义项、主题自定义项、业务指标自定义项）。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/design/customized-fields

表 8-1629 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 8-1630 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页查询条数, 即查询Y条数据。默认值50, 取值范围[1,100]。
offset	否	Integer	查询起始坐标, 即跳过X条数据, 仅支持0或limit的整数倍, 不满足则向下取整, 默认值0。

参数	是否必选	参数类型	描述
type	是	String	自定义项类型。 枚举值： <ul style="list-style-type: none"> <li>• TABLE: 表自定义项</li> <li>• ATTRIBUTE: 属性自定义项</li> <li>• SUBJECT: 主题自定义项</li> <li>• METRIC: 业务指标自定义项</li> </ul>

## 请求参数

表 8-1631 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1632 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1633 data

参数	参数类型	描述
value	value object	value, 统一的返回结果的外层数据结构。

表 8-1634 value

参数	参数类型	描述
total	Integer	总量。
records	Array of CustomizedFields VO objects	CustomizedFieldsVO信息。

表 8-1635 CustomizedFieldsVO

参数	参数类型	描述
id	String	编码, ID字符串。
name_ch	String	自定义项中文名称。
name_en	String	自定义项英文名称。
not_null	Boolean	是否必填。
optional_values	String	可选值。当可选值有多个时, 用分号分隔。
type	String	自定义项类型。 枚举值: <ul style="list-style-type: none"> <li>• TABLE: 表自定义项</li> <li>• ATTRIBUTE: 属性自定义项</li> <li>• SUBJECT: 主题自定义项</li> <li>• METRIC: 业务指标自定义项</li> </ul>
ordinal	Integer	系统排序字段, 新建、修改时不需要填写。
description	String	自定义项描述。
create_by	String	创建人, 只读。
update_by	String	更新人, 只读。



参数	参数类型	描述
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

**状态码：400**

表 8-1636 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1637 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1638 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

查询表的自定义项。

```
GET https://{endpoint}/v2/{project_id}/design/customized-fields?type=TABLE
```

## 响应示例

**状态码：200**

查询自定义项，返回结果是CustomizedFieldsVO集合。

```
{
  "data": {
    "value": {
      "total": 2,
      "records": [ {
        "id": "1209138753558437888",
        "name_ch": "aaa",
        "name_en": "aaa",
        "not_null": false,
        "optional_values": "",
        "type": "TABLE",
        "ordinal": 0,
        "description": "",
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "create_time": "2024-02-19T14:05:44+08:00",
        "update_time": "2024-03-14T15:23:50+08:00"
      }, {
        "id": "1209138753558437889",
        "name_ch": "bbb",
        "name_en": "bbb",
        "not_null": false,
        "optional_values": "",
        "type": "TABLE",
        "ordinal": 1,
        "description": "",
        "create_by": "test_uesr",
        "update_by": "test_uesr",
        "create_time": "2024-02-19T14:05:44+08:00",
        "update_time": "2024-03-14T15:23:50+08:00"
      }
    ]
  }
}
```

**状态码：400**

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchCustomizedFieldsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        SearchCustomizedFieldsRequest request = new SearchCustomizedFieldsRequest();
        try {
            SearchCustomizedFieldsResponse response = client.searchCustomizedFields(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchCustomizedFieldsRequest()
        response = client.search_customized_fields(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchCustomizedFieldsRequest{}
    response, err := client.SearchCustomizedFields(request)
    if err == nil {
```

```
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询自定义项，返回结果是CustomizedFieldsVO集合。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.24.2 修改自定义项

### 功能介绍

修改自定义项（包括表自定义项、属性自定义项、主题自定义项、业务指标自定义项）。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/design/customized-fields

表 8-1639 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 8-1640 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1641 请求 Body 参数

参数	是否必选	参数类型	描述
type	是	String	自定义项类型。 枚举值： <ul style="list-style-type: none"> <li>• TABLE: 表自定义项</li> <li>• ATTRIBUTE: 属性自定义项</li> <li>• SUBJECT: 主题自定义项</li> <li>• METRIC: 业务指标自定义项</li> </ul>
fields	否	Array of <a href="#">CustomizedFieldsVO</a> objects	自定义项列表。

表 8-1642 CustomizedFieldsVO

参数	是否必选	参数类型	描述
id	否	String	编码，ID字符串。

参数	是否必选	参数类型	描述
name_ch	是	String	自定义项中文名称。
name_en	是	String	自定义项英文名称。
not_null	是	Boolean	是否必填。
optional_values	否	String	可选值。当可选值有多个时，用分号分隔。
type	是	String	自定义项类型。 枚举值： <ul style="list-style-type: none"> <li>• TABLE: 表自定义项</li> <li>• ATTRIBUTE: 属性自定义项</li> <li>• SUBJECT: 主题自定义项</li> <li>• METRIC: 业务指标自定义项</li> </ul>
ordinal	否	Integer	系统排序字段，新建、修改时不需要填写。
description	否	String	自定义项描述。

## 响应参数

状态码：200

表 8-1643 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	data，统一的返回结果的最外层数据结构。

表 8-1644 data

参数	参数类型	描述
value	Array of <b>CustomizedFieldsVO</b> objects	数据连接信息数组

表 8-1645 CustomizedFieldsVO

参数	参数类型	描述
id	String	编码，ID字符串。

参数	参数类型	描述
name_ch	String	自定义项中文名称。
name_en	String	自定义项英文名称。
not_null	Boolean	是否必填。
optional_values	String	可选值。当可选值有多个时，用分号分隔。
type	String	自定义项类型。 枚举值： <ul style="list-style-type: none"> <li>• TABLE: 表自定义项</li> <li>• ATTRIBUTE: 属性自定义项</li> <li>• SUBJECT: 主题自定义项</li> <li>• METRIC: 业务指标自定义项</li> </ul>
ordinal	Integer	系统排序字段，新建、修改时不需要填写。
description	String	自定义项描述。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

**状态码：400**

表 8-1646 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**



表 8-1647 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1648 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

修改表的自定义项。

PUT https://{endpoint}/v2/{project\_id}/design/customized-fields

```
{
  "type": "TABLE",
  "fields": [ {
    "id": "1211611269321355264",
    "name_ch": "自定义项1",
    "name_en": "selfDefine1",
    "not_null": false,
    "optional_values": "",
    "description": "测试1",
    "ordinal": 0,
    "type": "TABLE"
  }, {
    "id": "1211611269321355265",
    "name_ch": "自定义项2",
    "name_en": "selfDefine2",
    "not_null": true,
    "optional_values": "",
    "description": "修改2",
    "ordinal": 1,
    "type": "TABLE"
  }
  ]
}
```

## 响应示例

状态码：200

Success

```
{
  "data": {
```

```

"value": [ {
  "id": "1211611269321355264",
  "name_ch": "自定义项1",
  "name_en": "selfDefine1",
  "not_null": false,
  "optional_values": "",
  "type": "TABLE",
  "ordinal": 0,
  "description": "测试1",
  "create_by": null,
  "update_by": null,
  "create_time": null,
  "update_time": null
}, {
  "id": "1211611269321355265",
  "name_ch": "自定义项2",
  "name_en": "selfDefine2",
  "not_null": true,
  "optional_values": "",
  "type": "TABLE",
  "ordinal": 1,
  "description": "修改2",
  "create_by": null,
  "update_by": null,
  "create_time": null,
  "update_time": null
}
]
}
}

```

**状态码: 400**

BadRequest

```

{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}

```

**状态码: 401**

Unauthorized

```

{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}

```

**状态码: 403**

Forbidden

```

{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

修改表的自定义项。

```

package com.huaweicloud.sdk.test;

```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ModifyCustomizedFieldsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ModifyCustomizedFieldsRequest request = new ModifyCustomizedFieldsRequest();
        CustomizedFieldsVOList body = new CustomizedFieldsVOList();
        List<CustomizedFieldsVO> listbodyFields = new ArrayList<>();
        listbodyFields.add(
            new CustomizedFieldsVO()
                .withId("1211611269321355264")
                .withNameCh("自定义项1")
                .withNameEn("selfDefine1")
                .withNotNull(false)
                .withOptionalValues("")
                .withType(CustomizedFieldsVO.TypeEnum.fromValue("TABLE"))
                .withOrdinal(0)
                .withDescription("测试1")
        );
        listbodyFields.add(
            new CustomizedFieldsVO()
                .withId("1211611269321355265")
                .withNameCh("自定义项2")
                .withNameEn("selfDefine2")
                .withNotNull(true)
                .withOptionalValues("")
                .withType(CustomizedFieldsVO.TypeEnum.fromValue("TABLE"))
                .withOrdinal(1)
                .withDescription("修改2")
        );
        body.withFields(listbodyFields);
        body.withType(CustomizedFieldsVOList.TypeEnum.fromValue("TABLE"));
        request.withBody(body);
        try {
            ModifyCustomizedFieldsResponse response = client.modifyCustomizedFields(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
```

```
e.printStackTrace();
System.out.println(e.getStatusCode());
System.out.println(e.getRequestId());
System.out.println(e.getErrorCode());
System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

修改表的自定义项。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ModifyCustomizedFieldsRequest()
        listFieldsbody = [
            CustomizedFieldsVO(
                id="1211611269321355264",
                name_ch="自定义项1",
                name_en="selfDefine1",
                not_null=False,
                optional_values="",
                type="TABLE",
                ordinal=0,
                description="测试1"
            ),
            CustomizedFieldsVO(
                id="1211611269321355265",
                name_ch="自定义项2",
                name_en="selfDefine2",
                not_null=True,
                optional_values="",
                type="TABLE",
                ordinal=1,
                description="修改2"
            )
        ]
        request.body = CustomizedFieldsVOList(
            fields=listFieldsbody,
            type="TABLE"
        )
        response = client.modify_customized_fields(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

修改表的自定义项。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ModifyCustomizedFieldsRequest{
        idFields:= "1211611269321355264"
        optionalValuesFields:= ""
        ordinalFields:= int32(0)
        descriptionFields:= "测试1"
        idFields1:= "1211611269321355265"
        optionalValuesFields1:= ""
        ordinalFields1:= int32(1)
        descriptionFields1:= "修改2"
        var listFieldsbody = []model.CustomizedFieldsVo{
            {
                Id: &idFields,
                NameCh: "自定义项1",
                NameEn: "selfDefine1",
                NotNull: false,
                OptionalValues: &optionalValuesFields,
                Type: model.GetCustomizedFieldsVoTypeEnum().TABLE,
                Ordinal: &ordinalFields,
                Description: &descriptionFields,
            },
            {
                Id: &idFields1,
                NameCh: "自定义项2",
                NameEn: "selfDefine2",
                NotNull: true,
                OptionalValues: &optionalValuesFields1,
                Type: model.GetCustomizedFieldsVoTypeEnum().TABLE,
                Ordinal: &ordinalFields1,
            }
        }
    }
```

```
        Description: &descriptionFields1,
    },
}
request.Body = &model.CustomizedFieldsVoList{
    Fields: &listFieldsbody,
    Type: model.GetCustomizedFieldsVoListTypeEnum().TABLE,
}
response, err := client.ModifyCustomizedFields(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.25 标签接口

### 8.25.1 添加标签

#### 功能介绍

根据资产（表或属性）的ID给资产打上标签。

#### 调用方法

请参见[如何调用API](#)。

#### URI

PUT /v2/{project\_id}/design/{entity\_id}/tags

表 8-1649 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
entity_id	是	String	表的ID，ID字符串。

表 8-1650 Query 参数

参数	是否必选	参数类型	描述
attr_id	否	String	属性的ID，ID字符串。
tags	是	Array of strings	标签名。

## 请求参数

表 8-1651 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1652 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回的数据信息。

表 8-1653 data

参数	参数类型	描述
value	String	给表/属性打标签/删除标签的接口返回，返回的value是null则表示成功。

**状态码：400**

表 8-1654 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

表 8-1655 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

表 8-1656 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。



## 请求示例

- 给ID为1217123755210469378的表打标签aaa、bbb。  
PUT https://{endpoint}/v2/{project\_id}/design/1217123755210469378/tags?tags=bbb&tags=aaa
- 给ID为1217123755210469378的表中ID为1217123755210469379的属性打标签aaa、bbb。  
PUT https://{endpoint}/v2/{project\_id}/design/1217123755210469378/tags?attr\_id=1217123755210469379&tags=aaa&tags=bbb

## 响应示例

### 状态码：200

Success, 返回data数据是否成功。

```
{
  "data" : {
    "value" : null
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class AddDesignEntityTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        AddDesignEntityTagsRequest request = new AddDesignEntityTagsRequest();
        request.withEntityId("{entity_id}");
        try {
            AddDesignEntityTagsResponse response = client.addDesignEntityTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = AddDesignEntityTagsRequest()
    request.entity_id = "{entity_id}"
    response = client.add_design_entity_tags(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddDesignEntityTagsRequest{}
    request.EntityId = "{entity_id}"
    response, err := client.AddDesignEntityTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是否成功。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.25.2 删除标签

### 功能介绍

根据资产（表或属性）的ID删除资产标签。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/{entity\_id}/tags

表 8-1657 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
entity_id	是	String	表的ID, ID字符串。

表 8-1658 Query 参数

参数	是否必选	参数类型	描述
attr_id	否	String	属性的ID, ID字符串。
tag	是	String	标签名。

## 请求参数

表 8-1659 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1660 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	返回的数据信息。

表 8-1661 data

参数	参数类型	描述
value	String	给表/属性打标签/删除标签的接口返回，返回的value是null则表示成功。

状态码：400

表 8-1662 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1663 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1664 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

- 将标签ddd从ID为1217123755210469378的表上移除。

```
DELETE https://{endpoint}/v2/{project_id}/design/1217123755210469378/tags?tag=ddd
```
- 将标签ccc从ID为1217123755210469378的表中ID为1217123755214663680的属性上移除。

```
DELETE https://{endpoint}/v2/{project_id}/design/1217123755210469378/tags?attr_id=1217123755214663680&tag=ccc
```

## 响应示例

状态码：200

Success，返回data数据是否成功。

```
{
  "data" : {
```

```
"value" : null
}
}
```

**状态码：400**

BadRequest

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user request is illegal."
}
```

**状态码：401**

Unauthorized

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "User authentication failed."
}
```

**状态码：403**

Forbidden

```
{
  "error_code" : "DS.60xx",
  "error_msg" : "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class RemoveDesignEntityTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
RemoveDesignEntityTagsRequest request = new RemoveDesignEntityTagsRequest();
request.withEntityId("{entity_id}");
try {
    RemoveDesignEntityTagsResponse response = client.removeDesignEntityTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RemoveDesignEntityTagsRequest()
        request.entity_id = "{entity_id}"
        response = client.remove_design_entity_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
```



```

)
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RemoveDesignEntityTagsRequest{}
    request.EntityId = "{entity_id}"
    response, err := client.RemoveDesignEntityTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回data数据是否成功。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.26 质量规则接口

### 8.26.1 更新表的异常数据输出配置

#### 功能介绍

更新表的异常数据输出配置，包括是否生成异常数据、设置异常数据数据库或Schema、设置异常表表前缀/表后缀。

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v2/{project\_id}/design/{table\_id}/qualities

表 8-1665 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
table_id	是	String	表的ID，ID字符串。

## 请求参数

表 8-1666 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

表 8-1667 请求 Body 参数

参数	是否必选	参数类型	描述
biz_type	否	String	表类型。 枚举值： <ul style="list-style-type: none"> <li>TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>FACT_LOGIC_TABLE: 事实表</li> <li>DIMENSION: 维度</li> <li>DIMENSION_LOGIC_TABLE: 维度表</li> </ul>
dirty_out_switch	否	Boolean	异常数据输出开关。
dirty_out_database	否	String	异常数据输出库。
dirty_out_prefix	否	String	异常表前缀。
dirty_out_suffix	否	String	异常表后缀。

## 响应参数

状态码：200

表 8-1668 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	更新表的异常数据输出配置返回数据。

表 8-1669 data

参数	参数类型	描述
value	<b>TableModelVO</b> object	表的详情信息。

表 8-1670 TableModelVO

参数	参数类型	描述
id	String	编码，ID字符串。
model_id	String	所属关系建模的模型ID，ID字符串。
parent_table_id	String	父表ID，ID字符串。
parent_table_name	String	父表名称，只读。
parent_table_code	String	父表编码，只读。
related_logic_table_id	String	关联逻辑实体的ID，ID字符串。
related_logic_table_name	String	关联逻辑实体的名称。
related_logic_table_model_id	String	关联逻辑实体的模型ID，ID字符串。
related_logic_table_model_name	String	关联逻辑实体的模型名称。
model	<a href="#">WorkspaceVO</a> object	模型信息。
data_format	String	数据格式。
obs_bucket	String	obs桶。
obs_location	String	外表路径
configs	String	其他配置。
table_type	String	表类型，只读。
owner	String	负责人。
tb_name	String	表名。
dw_id	String	数据连接ID。
db_name	String	数据库名。
queue_name	String	dli数据连接执行sql所需的队列，数据连接类型为DLI时必须。
schema	String	DWS类型需要。
extend_info	String	扩展信息。
tb_guid	String	表物化后的guid，只读。
tb_id	String	数据表ID，只读。
logic_tb_name	String	逻辑实体名。

参数	参数类型	描述
logic_tb_guid	String	逻辑实体的guid，只读。
description	String	描述。
status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
logic_tb_id	String	逻辑实体的ID，ID字符串。
biz_catalog_id	String	归属的业务分类的id，ID字符串。
catalog_path	String	归属的业务分类的路径 { "l1Id": "", "l2Id": "", "l3Id": "" }。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如 1970-01-01T00:00:00Z。
tags	Array of <b>TagRecordVO</b> objects	表标签，只读。
approval_info	<b>ApprovalVO</b> object	审批信息,只读参数。业务对象最近一次的审批信息，包括审批的业务详情、审核人信息、审核时间等。
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。
attributes	Array of <b>TableModelAttributeVO</b> objects	表属性信息。

参数	参数类型	描述
mappings	Array of <a href="#">TableMappingVO</a> objects	表映射信息。
relations	Array of <a href="#">RelationVO</a> objects	关系。
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
dw_name	String	数据连接名称，只读，创建和更新时无需填写。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
l1_id	String	主题域分组ID，只读，ID字符串。
l2_id	String	主题域ID，只读，创建和更新时无需填写。
l3_id	String	业务对象ID，只读，ID字符串。
partition_conf	String	分区表达式。
dlf_task_id	String	DLF作业ID。
use_recently_partition	Boolean	是否使用最新分区。
reversed	Boolean	是否是逆向的。
dirty_out_switch	Boolean	异常数据输出开关。
dirty_out_database	String	异常数据输出库。
dirty_out_prefix	String	异常表前缀。
dirty_out_suffix	String	异常表后缀。
quality_owner	String	质量责任人。
quality_id	String	质量ID，ID字符串。

参数	参数类型	描述
distribute	String	DISTRIBUTE BY [HASH(column) REPLICATION]。 枚举值： <ul style="list-style-type: none"> <li>• HASH: 对指定的列进行Hash，通过映射，把数据分布到指定DN</li> <li>• REPLICATION: 表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据</li> </ul>
distribute_column	String	DISTRIBUTE BY HASH column.
is_partition	Boolean	是否分区表，只读。
physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_physical_table	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>

参数	参数类型	描述
technical_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
business_asset	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
meta_data_link	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>



参数	参数类型	描述
data_quality	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
summary_status	String	同步状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_NEED: 未同步</li> <li>• CREATE_SUCCESS: 创建成功</li> <li>• CREATE_FAILED: 创建失败</li> <li>• UPDATE_SUCCESS: 更新成功</li> <li>• UPDATE_FAILED: 更新失败</li> <li>• SUMMARY_SUCCESS: 整体成功</li> <li>• SUMMARY_FAILED: 整体失败</li> <li>• RUNNING: 整体运行</li> <li>• OFFLINE: 已下线</li> </ul>
dev_version	String	开发环境版本，只读，ID字符串。
prod_version	String	生产环境版本，只读，ID字符串。
dev_version_name	String	开发环境版本名称，只读。。
prod_version_name	String	生产环境版本名称，只读。。
env_type	String	开发生产环境类型。 枚举值： <ul style="list-style-type: none"> <li>• INVALID_TYPE: 无效环境</li> <li>• DEV_TYPE: 开发环境</li> <li>• PROD_TYPE: 生产环境</li> <li>• DEV_PROD_TYPE: 开发生产环境</li> </ul>
alias	String	别名。

参数	参数类型	描述
self_defined_fields	Array of <b>SelfDefinedField VO</b> objects	自定义项。
code	String	编码
has_related_physical_table	Boolean	是否存在关联物理表。
has_related_logic_table	Boolean	是否存在关联逻辑实体。

表 8-1671 WorkspaceVO

参数	参数类型	描述
id	String	编号，ID字符串。
name	String	工作区名字。
description	String	描述。
is_physical	Boolean	是否为物理表。
frequent	Boolean	是否为常用。
top	Boolean	分层治理。
level	String	数据治理分层。 枚举值： <ul style="list-style-type: none"> <li>SDI: 贴源数据层</li> <li>DWI: 数据整合层</li> <li>DWR: 数据报告层</li> <li>DM: 数据集市层</li> </ul>
dw_type	String	数据连接类型，对应表所在的数仓类型，取值可以为DWS、MRS_HIVE、POSTGRESQL、MRS_SPARK、CLICKHOUSE、MYSQL、ORACLE和DORIS等。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
create_by	String	创建人。
update_by	String	更新人。
type	String	工作区类型枚举。 枚举值： <ul style="list-style-type: none"> <li>• THIRD_NF: 关系建模</li> <li>• DIMENSION: 维度建模</li> </ul>
biz_catalog_ids	String	关联的业务分层的ID列表 {"l1Ids": [], "l2Ids": [], "l3Ids": []}。
databases	Array of strings	数据库名称数组。
table_model_prefix	String	模型校验前缀，长度不超过100，数字字母下划线组成，字母开头

表 8-1672 TagRecordVO

参数	参数类型	描述
id	String	编码，ID字符串。
tag_id	String	标签ID，ID字符串。
tag_name	String	标签名称。
biz_id	String	实体ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1673 ApprovalVO

参数	参数类型	描述
id	String	审批单ID，ID字符串。
tenant_id	String	项目ID，获取方式参考接口路径参数“project_id”。
name_ch	String	业务中文名。
name_en	String	业务英文名。
biz_id	String	业务ID，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_info	String	序列化之后的业务详情，类型是string。
biz_info_obj	Object	业务详情，类型是object。
biz_version	Integer	业务版本。
biz_status	String	实体的发布状态，只读，创建和更新时无需填写。 枚举值： <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
approval_status	String	业务审批状态，只读。 枚举值： <ul style="list-style-type: none"> <li>• DEVELOPING: 审核中</li> <li>• APPROVED: 审核通过</li> <li>• REJECT: 审核驳回</li> <li>• WITHDREW: 审核撤销</li> </ul>
approval_type	String	业务审核类型。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布</li> <li>• OFFLINE: 下线</li> </ul>
submit_time	String	提交时间。

参数	参数类型	描述
create_by	String	创建者。
l1	String	主题域分组中文名，只读，创建和更新时无需填写。
l2	String	主题域中文名，只读，创建和更新时无需填写。
l3	String	业务对象中文名，只读，创建和更新时无需填写。
approval_time	String	审核时间。
approver	String	审核人。
email	String	审核人邮箱。
msg	String	审核信息。
directory_path	String	目录树。

表 8-1674 TableModelAttributeVO

参数	参数类型	描述
id	String	编码，ID字符串。
name_en	String	字段名。
name_ch	String	业务属性。
description	String	描述。
obs_location	String	obs路径，子路径。
create_by	String	创建人。
update_by	String	更新人。
data_type	String	字段类型。
domain_type	String	字段类型所属域。 枚举值： <ul style="list-style-type: none"> <li>NUMBER: 数字类型</li> <li>STRING: 字符类型</li> <li>DATETIME: 日期类型</li> <li>BLOB: 大对象(BLOB)</li> <li>OTHER: 其它类型</li> </ul>
data_type_extend	String	数据类型扩展字段。
is_primary_key	Boolean	是否主键。



参数	参数类型	描述
is_partition_key	Boolean	是否分区键。
is_foreign_key	Boolean	是否外键。
extend_field	Boolean	是否继承的属性。
not_null	Boolean	是否不为空。
ordinal	Integer	序号。
table_model_id	String	所属关系建模的模型ID，ID字符串。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
tags	Array of <b>TagVO</b> objects	表标签，只读。
secrecy_levels	Array of <b>SecrecyLevelVO</b> objects	密级
stand_row_id	String	关联的数据标准的ID，ID字符串。
stand_row_name	String	关联的数据标准名称，只读。
quality_infos	Array of <b>QualityInfoVO</b> objects	质量信息，只读。
alias	String	别名。
self_defined_fields	Array of <b>SelfDefinedFieldVO</b> objects	自定义项。
code	String	逻辑属性编码
related_logic_attr_id	String	关联逻辑属性ID，ID字符串。
related_logic_attr_name	String	关联逻辑实体属性中文名称
related_logic_attr_name_en	String	关联逻辑实体属性英文名称

表 8-1675 TagVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	标签名。
description	String	描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
new_biz	<b>BizVersionManag</b> <b>eVO</b> object	业务版本管理，只读。

表 8-1676 SecrecyLevelVO

参数	参数类型	描述
id	String	密级ID，ID字符串。
uuid	String	数据安全主键。
name	String	密级名。
slevel	Integer	密级等级。
description	String	密级描述。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

参数	参数类型	描述
new_biz	<b>BizVersionManageVO</b> object	业务版本管理，只读。

表 8-1677 BizVersionManageVO

参数	参数类型	描述
id	String	ID信息，ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
biz_id	String	业务ID，ID字符串。
biz_info	String	业务对象信息。
status	String	<p>实体的发布状态，只读，创建和更新时无需填写。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• DRAFT: 草稿</li> <li>• PUBLISH_DEVELOPING: 发布待审核</li> <li>• PUBLISHED: 已发布</li> <li>• OFFLINE_DEVELOPING: 下线待审核</li> <li>• OFFLINE: 已下线</li> <li>• REJECT: 已驳回</li> </ul>
biz_version	Integer	业务版本，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1678 QualityInfoVO

参数	参数类型	描述
id	String	编码ID, ID字符串。
table_id	String	表ID, 只读, ID字符串。
attr_id	String	属性ID, 只读, ID字符串。

参数	参数类型	描述
biz_type	String	<p>业务实体类型。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• AGGREGATION_LOGIC_TABLE: 汇总表</li> <li>• ATOMIC_INDEX: 原子指标</li> <li>• ATOMIC_METRIC: 原子指标（新）</li> <li>• BIZ_CATALOG: 流程架构目录</li> <li>• BIZ_METRIC: 业务指标</li> <li>• CODE_TABLE: 码表</li> <li>• COMMON_CONDITION: 通用限定</li> <li>• COMPOSITE_METRIC: 复合指标（新）</li> <li>• COMPOUND_METRIC: 复合指标</li> <li>• CONDITION_GROUP: 限定分组</li> <li>• DEGENERATE_DIMENSION: 退化维度</li> <li>• DERIVATIVE_INDEX: 衍生指标</li> <li>• DERIVED_METRIC: 衍生指标（新）</li> <li>• DIMENSION: 维度</li> <li>• DIMENSION_ATTRIBUTE: 维度属性</li> <li>• DIMENSION_HIERARCHIES: 维度层级</li> <li>• DIMENSION_LOGIC_TABLE: 维度表</li> <li>• DIMENSION_TABLE_ATTRIBUTE: 维度属性</li> <li>• DIRECTORY: 目录</li> <li>• FACT_ATTRIBUTE: 事实表属性</li> <li>• FACT_DIMENSION: 事实表维度</li> <li>• FACT_LOGIC_TABLE: 事实表</li> <li>• FACT_MEASURE: 事实表度量</li> <li>• FUNCTION: 函数</li> <li>• INFO_ARCH: 信息架构（批量修改主题使用）</li> <li>• MODEL: 模型</li> <li>• QUALITY_RULE: 质量规则</li> <li>• SECRECY_LEVEL: 密级</li> <li>• STANDARD_ELEMENT: 数据标准</li> <li>• STANDARD_ELEMENT_TEMPLATE: 数据标准模板</li> </ul>

参数	参数类型	描述
		<ul style="list-style-type: none"> <li>• SUBJECT: 主题</li> <li>• SUMMARY_DIMENSION_ATTRIBUTE: 汇总表维度属性</li> <li>• SUMMARY_INDEX: 汇总表指标属性</li> <li>• SUMMARY_TIME: 汇总表时间周期属性</li> <li>• TABLE_MODEL: 关系模型（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_ATTRIBUTE: 关系模型属性（逻辑模型/物理模型）</li> <li>• TABLE_MODEL_LOGIC: 逻辑实体</li> <li>• TABLE_TYPE: 表类型</li> <li>• TAG: 标签</li> <li>• TIME_CONDITION: 时间限定</li> </ul>
data_quality_id	String	质量ID，ID字符串。
show_control	Integer	是否要显示正则表达式。
data_quality_name	String	质量名称。
alert_conf	String	告警配置。
expression	String	正则相关校验规则中正则配置。
extend_info	String	扩展信息。
from_standard	Boolean	是否来源于数据标准质量配置，只读。
result_description	String	结果说明。
create_by	String	创建人，只读。
update_by	String	更新人，只读。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。



表 8-1679 TableMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
name	String	名称。
description	String	描述。
target_table_id	String	目的表ID，ID字符串。
src_model_id	String	来源表在关系建模中的模型ID，ID字符串。
src_model_name	String	来源模型名称。
view_text	String	采集的视图来源，dws视图逆向使用。
target_table_name	String	目的表名称。
details	Array of <a href="#">TableMappingDetailVO</a> objects	详情。
source_tables	Array of <a href="#">MappingSourceTableVO</a> objects	映射的表信息。
source_fields	Array of <a href="#">MappingSourceFieldVO</a> objects	映射的字段信息。
create_time	String	创建时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。

表 8-1680 TableMappingDetailVO

参数	参数类型	描述
id	String	编码，ID字符串。
mapping_id	String	名称。

参数	参数类型	描述
target_attr_id	String	目的字段ID, ID字符串。
target_attr_name	String	目的字段排序。
src_table_ids	String	源表ID。
src_table_names	Array of strings	源表名称数组, 只读。
src_table_db_names	Array of strings	源表db名称数组, 只读。
src_table_model_ids	Array of strings	源表在关系建模中的模型ID数组, 只读, ID字符串。
src_table_id_list	Array of strings	源表ID数组, 只读, ID字符串。
src_attr_ids	String	源表字段ID。
src_attr_names	Array of strings	源表字段名称数组, 只读。
src_attr_id_list	Array of strings	源表字段ID数组, 只读, ID字符串。
remark	String	备注。
create_time	String	创建时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
update_time	String	更新时间, 只读, 格式遵循RFC3339, 精确到秒, UTC时区, 即yyyy-mm-ddTHH:MM:SSZ, 如1970-01-01T00:00:00Z。
create_by	String	创建人。
update_by	String	更新人。
target_attr	Object	目标属性。

表 8-1681 MappingSourceTableVO

参数	参数类型	描述
table1_id	String	表1ID, ID字符串。
table2_id	String	表2ID, ID字符串。
table1_name	String	表1名称。
table2_name	String	表2名称。

参数	参数类型	描述
join_type	String	关联类型。 枚举值： <ul style="list-style-type: none"> <li>• LEFT: 左外连接</li> <li>• RIGHT: 右外连接</li> <li>• INNER: 内连接</li> <li>• FULL: 全连接</li> </ul>
join_fields	Array of <a href="#">MappingJoinFieldVO</a> objects	on条件。

表 8-1682 MappingJoinFieldVO

参数	参数类型	描述
field1_id	String	属性1ID, ID字符串。
field2_id	String	属性2ID, ID字符串。
field1_name	String	属性1名称。
field2_name	String	属性2名称。

表 8-1683 MappingSourceFieldVO

参数	参数类型	描述
target_field_id	String	目标字段ID, 当前表的某个字段, ID字符串。
target_field_name	String	目标字段编码。
field_ids	String	来源字段ID, 多个ID以逗号分隔。
transform_expression	String	转换表达式。
field_names	Array of strings	来源字段名称列表。
changed	Boolean	字段是否发生变化。

表 8-1684 RelationVO

参数	参数类型	描述
id	String	编码, ID字符串。

参数	参数类型	描述
source_table_id	String	源表ID，ID字符串。
target_table_id	String	目标表ID，ID字符串。
name	String	关系名称。
source_table_name	String	源表名称。
target_table_name	String	目的表名称。
role	String	角色。
tenant_id	String	租户ID。
source_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
target_type	String	关系类型，只读。 枚举值： <ul style="list-style-type: none"> <li>ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中有且只有一条数据与之对应。</li> <li>ZERO_OR_ONE: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中最多有一条数据与之对应。</li> <li>ZERO_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中可能有多条数据与之对应。</li> <li>ONE_OR_N: 表示每条子（父）逻辑实体数据在父（子）逻辑实体中至少有一条数据与之对应。</li> </ul>
create_by	String	创建人。
update_by	String	更新人。

参数	参数类型	描述
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
mappings	Array of <b>RelationMappingVO</b> objects	表属性信息，只读。

表 8-1685 RelationMappingVO

参数	参数类型	描述
id	String	编码，ID字符串。
relation_id	String	关系ID，ID字符串。
source_field_id	String	源字段ID，ID字符串。
target_field_id	String	目标字段ID，ID字符串。
source_field_name	String	源表名称。
target_field_name	String	目的表名称。
create_by	String	创建人。
update_by	String	更新人。
create_time	String	创建时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。
update_time	String	更新时间，只读，格式遵循RFC3339，精确到秒，UTC时区，即yyyy-mm-ddTHH:MM:SSZ，如1970-01-01T00:00:00Z。

表 8-1686 SelfDefinedFieldVO

参数	参数类型	描述
fd_name_ch	String	自定义项中文名。
fd_name_en	String	自定义项英文名。

参数	参数类型	描述
not_null	Boolean	是否必填。
fd_value	String	属性值。

**状态码：400**

**表 8-1687** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：401**

**表 8-1688** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码：403**

**表 8-1689** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

根据请求体参数更新表级质量规则。

```
PUT https://{endpoint}/v2/{project_id}/design/1217123755210469376/qualities
{
```

```
"biz_type": "TABLE_MODEL",
"dirty_out_switch": true,
"dirty_out_database": "default",
"dirty_out_prefix": "err_",
"dirty_out_suffix": "_2020_02_28"
}
```

## 响应示例

**状态码：200**

Success, 返回更新后表的全部信息。

```
{
  "data": {
    "value": {
      "id": "1217123755210469376",
      "model_id": "873242752312385536",
      "parent_table_id": null,
      "parent_table_name": null,
      "parent_table_code": null,
      "model": {
        "name": "test",
        "type": "THIRD_NF"
      },
      "data_format": "Parquet",
      "obs_bucket": null,
      "obs_location": null,
      "configs": "{}",
      "table_type": "MANAGED",
      "owner": "",
      "tb_name": "staff_info",
      "dw_id": "2c9080f07b03db8d017b198a4a20002f",
      "db_name": "ctz_test",
      "queue_name": "default",
      "schema": "",
      "extend_info": "{\"dirtyOutDatabase\":\"\",\"dirtyOutPrefix\":\"\",\"dirtyOutSuffix\":\"\",\"dirtyOutSwitch\":\"false\"}",
      "tb_guid": null,
      "tb_id": "NativeTable-2c9080f07b03db8d017b198a4a20002f-ctz_test-staff_info",
      "logic_tb_name": "staff_info",
      "logic_tb_guid": "1ab1a58b-9412-4901-bb80-4ccaaa0580e4",
      "description": "无",
      "status": "PUBLISHED",
      "logic_tb_id": null,
      "biz_catalog_id": "1012306178602041344",
      "catalog_path": "工厂/人员管理/人员档案",
      "create_by": "abc",
      "update_by": "abc",
      "create_time": "2022-08-25T10:55:00+08:00",
      "update_time": "2022-08-25T10:55:45+08:00",
      "approval_info": {
        "id": "1012314318533988352",
        "tenant_id": null,
        "name_ch": null,
        "name_en": null,
        "biz_id": "1012314146584301568",
        "biz_type": null,
        "biz_info": null,
        "biz_info_obj": null,
        "biz_version": 0,
        "biz_status": "PUBLISHED",
        "approval_status": "APPROVED",
        "approval_type": null,
        "submit_time": "2022-08-25T10:55:41+08:00",
        "create_by": null,
        "l1": null,
        "l2": null,
        "l3": null,
      }
    }
  }
}
```

```

"approval_time": "2022-08-25T10:55:41+08:00",
"approver": "abc",
"email": null,
"msg": "Fast Approval OK.",
"directory_path": null
},
"new_biz": null,
"attributes": [ {
  "id": "1012314146626244608",
  "name_en": "id",
  "name_ch": "主键",
  "description": "",
  "obs_location": null,
  "create_by": "abc",
  "update_by": "abc",
  "data_type": "STRING",
  "domain_type": "STRING",
  "data_type_extend": "",
  "is_primary_key": true,
  "is_partition_key": false,
  "is_foreign_key": false,
  "extend_field": false,
  "not_null": true,
  "ordinal": 1,
  "table_model_id": null,
  "create_time": "2022-08-25T10:55:00+08:00",
  "update_time": "2022-08-25T10:55:00+08:00",
  "tags": [ {
    "name": 123
  } ],
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": [ {
    "data_quality_id": 3
  } ],
  "alias": "",
  "self_defined_fields": [ ]
}, {
  "id": "1012314146626244609",
  "name_en": "name",
  "name_ch": "名称",
  "description": "",
  "obs_location": null,
  "create_by": "abc",
  "update_by": "abc",
  "data_type": "STRING",
  "domain_type": "STRING",
  "data_type_extend": "",
  "is_primary_key": false,
  "is_partition_key": false,
  "is_foreign_key": false,
  "extend_field": false,
  "not_null": false,
  "ordinal": 2,
  "table_model_id": null,
  "create_time": "2022-08-25T10:55:00+08:00",
  "update_time": "2022-08-25T10:55:00+08:00",
  "tags": [ {
    "name": 123
  } ],
  "stand_row_id": null,
  "stand_row_name": null,
  "quality_infos": [ {
    "data_quality_id": 3
  } ],
  "alias": "",
  "self_defined_fields": [ ]
}, {
  "id": "1012314146592690176",

```



```

"name_en": "age",
"name_ch": "年龄",
"description": "",
"obs_location": null,
"create_by": "abc",
"update_by": "abc",
"data_type": "STRING",
"domain_type": "STRING",
"data_type_extend": "",
"is_primary_key": false,
"is_partition_key": false,
"is_foreign_key": false,
"extend_field": false,
"not_null": false,
"ordinal": 3,
"table_model_id": null,
"create_time": "2022-08-25T10:55:00+08:00",
"update_time": "2022-08-25T10:55:00+08:00",
"tags": [ {
  "name": 123
} ],
"stand_row_id": null,
"stand_row_name": null,
"quality_infos": [ {
  "data_quality_id": 3
} ],
"alias": "",
"self_defined_fields": [ ]
} ],
"mappings": [ {
  "name": "test_mapping"
} ],
"relations": [ {
  "name": "test_relation"
} ],
"dw_type": "DWS",
"dw_name": null,
"l1": "工厂",
"l2": "人员管理",
"l3": "人员档案",
"l1_id": "1012305751919689728",
"l2_id": "1012306048792526848",
"l3_id": "1012306178602041344",
"partition_conf": null,
"dlf_task_id": null,
"use_recently_partition": false,
"reversed": null,
"dirty_out_switch": false,
"dirty_out_database": "",
"dirty_out_prefix": "",
"dirty_out_suffix": "",
"quality_owner": null,
"quality_id": "1012314332484304897",
"distribute": null,
"distribute_column": null,
"is_partition": false,
"physical_table": "CREATE_SUCCESS",
"technical_asset": "CREATE_FAILED",
"business_asset": "UPDATE_SUCCESS",
"meta_data_link": "UPDATE_FAILED",
"data_quality": "CREATE_SUCCESS",
"summary_status": "SUMMARY_FAILED",
"alias": null,
"self_defined_fields": null
}
}
}

```

状态码: 400

### BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

### Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

### Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据请求体参数更新表级质量规则。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateDesignTableQualitySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateDesignTableQualityRequest request = new UpdateDesignTableQualityRequest();
```

```
request.withTableId("{table_id}");
UpdateDesignTableQualityParam body = new UpdateDesignTableQualityParam();
body.withDirtyOutSuffix("_2020_02_28");
body.withDirtyOutPrefix("err_");
body.withDirtyOutDatabase("default");
body.withDirtyOutSwitch(true);
body.withBizType(UpdateDesignTableQualityParam.BizTypeEnum.fromValue("TABLE_MODEL"));
request.withBody(body);
try {
    UpdateDesignTableQualityResponse response = client.updateDesignTableQuality(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

根据请求体参数更新表级质量规则。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateDesignTableQualityRequest()
        request.table_id = "{table_id}"
        request.body = UpdateDesignTableQualityParam(
            dirty_out_suffix="_2020_02_28",
            dirty_out_prefix="err_",
            dirty_out_database="default",
            dirty_out_switch=True,
            biz_type="TABLE_MODEL"
        )
        response = client.update_design_table_quality(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

根据请求体参数更新表级质量规则。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDesignTableQualityRequest{}
    request.TableId = "{table_id}"
    dirtyOutSuffixUpdateDesignTableQualityParam:= "_2020_02_28"
    dirtyOutPrefixUpdateDesignTableQualityParam:= "err_"
    dirtyOutDatabaseUpdateDesignTableQualityParam:= "default"
    dirtyOutSwitchUpdateDesignTableQualityParam:= true
    bizTypeUpdateDesignTableQualityParam:=
model.GetUpdateDesignTableQualityParamBizTypeEnum().TABLE_MODEL
    request.Body = &model.UpdateDesignTableQualityParam{
        DirtyOutSuffix: &dirtyOutSuffixUpdateDesignTableQualityParam,
        DirtyOutPrefix: &dirtyOutPrefixUpdateDesignTableQualityParam,
        DirtyOutDatabase: &dirtyOutDatabaseUpdateDesignTableQualityParam,
        DirtyOutSwitch: &dirtyOutSwitchUpdateDesignTableQualityParam,
        BizType: &bizTypeUpdateDesignTableQualityParam,
    }
    response, err := client.UpdateDesignTableQuality(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回更新后表的全部信息。
400	BadRequest
401	Unauthorized
403	Forbidden

## 8.26.2 清空质量规则

### 功能介绍

清空表的质量规则。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/design/{table\_id}/qualities

表 8-1690 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
table_id	是	String	表的ID, ID字符串。

表 8-1691 Query 参数

参数	是否必选	参数类型	描述
table_type	是	String	表类型，默认值是业务表。 TABLE_MODEL(业务表(逻辑实体/物理表))、 AGGREGATION_LOGIC_TABLE(汇总表)、 FACT_LOGIC_TABLE(事实表)、 DIMENSION_LOGIC_TABLE(维度表)。 <ul style="list-style-type: none"> <li>• TABLE_MODEL</li> <li>• AGGREGATION_LOGIC_TABLE</li> <li>• FACT_LOGIC_TABLE</li> <li>• DIMENSION_LOGIC_TABLE</li> </ul>

## 请求参数

表 8-1692 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值） 使用Token认证时必选。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Project-Id	否	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 多project场景采用AK/SK认证的接口请求，则该字段必选。
Content-Type	否	String	默认值：application/json;charset=UTF-8 可选，有Body体的情况下必选，没有Body体则无需填写和校验。

## 响应参数

状态码：200

表 8-1693 响应 Body 参数

参数	参数类型	描述
data	<b>data</b> object	清空质量规则返回数据。

表 8-1694 data

参数	参数类型	描述
value	Boolean	是否清除成功。

状态码：400

表 8-1695 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：401

表 8-1696 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码：403

表 8-1697 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DS.6000，请求处理失败。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

将ID为1217123755210469378、表类型为TABLE\_MODEL的表的质量规则清空。

```
DELETE https://{endpoint}/v2/{project_id}/design/1217123755210469378/qualities?
table_type=TABLE_MODEL
```

## 响应示例

### 状态码：200

Success, 返回是否删除成功。

```
{
  "data": {
    "value": true
  }
}
```

### 状态码：400

BadRequest

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user request is illegal."
}
```

### 状态码：401

Unauthorized

```
{
  "error_code": "DS.60xx",
  "error_msg": "User authentication failed."
}
```

### 状态码：403

Forbidden

```
{
  "error_code": "DS.60xx",
  "error_msg": "The user does not have permission to call this API."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class RemoveDesignQualityInfosSolution {
```



```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    RemoveDesignQualityInfosRequest request = new RemoveDesignQualityInfosRequest();
    request.withTableId("{table_id}");
    try {
        RemoveDesignQualityInfosResponse response = client.removeDesignQualityInfos(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RemoveDesignQualityInfosRequest()
        request.table_id = "{table_id}"
        response = client.remove_design_quality_infos(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RemoveDesignQualityInfosRequest{}
    request.TableId = "{table_id}"
    response, err := client.RemoveDesignQualityInfos(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回是否删除成功。
400	BadRequest
401	Unauthorized

状态码	描述
403	Forbidden

# 9 数据质量 API

## 9.1 目录接口

### 9.1.1 获取作业目录

#### 功能介绍

获取作业目录。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/quality/categories

表 9-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-2 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	分页时每页的条数，最大值为100。
offset	否	Integer	分页偏移量。

## 请求参数

表 9-3 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
accept	是	String	application/json
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-4 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of <a href="#">CategoryDetailVO</a> objects	自定义项列表。

表 9-5 CategoryDetailVO

参数	参数类型	描述
id	Long	目录ID，根目录的ID为0。
name	String	名称。
type	String	类型：built_in表示系统内置，user-defined表示用户自定义。
pid	String	父目录ID。
sub_categories	Array of <a href="#">SubCategoryDetailVO</a> objects	子目录。

表 9-6 SubCategoryDetailVO

参数	参数类型	描述
id	Long	目录ID，根目录的ID为0。

参数	参数类型	描述
name	String	名称。
type	String	类型：built_in表示系统内置，user-defined表示用户自定义。
pid	String	父目录ID。
sub_categories	Object	子目录。

**状态码： 400**

**表 9-7** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

**状态码： 500**

**表 9-8** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

**状态码： 200**

Success

```
[ {
  "name" : "全部",
  "id" : 0,
  "type" : "built_in",
  "pid" : "-1",
  "sub_categories" : [ {
    "name" : "qwe",
    "id" : 1012384968858652700,
    "type" : "user-defined",
```

```
"pid" : "0",
"sub_categories" : [ {
  "id" : 1,
  "name" : "name"
} ]
} ]
} ]
} ]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListCategorySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCategoryRequest request = new ListCategoryRequest();
        try {
            ListCategoryResponse response = client.listCategory(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCategoryRequest()
        response = client.list_category(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```



```

Build()

request := &model.ListCategoryRequest{}
response, err := client.ListCategory(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.2 规则模板接口

### 9.2.1 获取规则模板列表

#### 功能介绍

分页获取规则模板列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/quality/rule-templates

表 9-9 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-10 Query 参数

参数	是否必选	参数类型	描述
category_id	否	Long	目录ID。
name	否	String	规则模板名称。
system_template	否	Boolean	是否只查询系统模板。
creator	否	String	创建者。
limit	否	Integer	分页时每页的条数，最大值为100。
offset	否	Integer	分页偏移量。

## 请求参数

表 9-11 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-12 响应 Body 参数

参数	参数类型	描述
count	Long	总条数。
resources	Array of <a href="#">RuleTemplateDetailVO</a> objects	分页数据。

表 9-13 RuleTemplateDetailVO

参数	参数类型	描述
id	Long	规则模板ID。

参数	参数类型	描述
name	String	规则模板名称。
category_id	Long	目录ID
dimension	String	维度，Completeness表示完整性，Uniqueness表示唯一性，Timeliness表示及时性，Validity表示有效性，Accuracy表示准确性，Consistency表示一致性。
type	String	规则类型，Field表示字段级规则，Table表示表级规则，Database表示库级规则，Cross-field表示跨字段级规则，Customize表示自定义规则。
system_template	Boolean	是否为系统模板。
sql_info	String	定义关系。
abnormal_table_template	String	异常表模板。
result_description	String	结果说明。
create_time	Long	创建时间，13位时间戳（精确到毫秒）。
creator	String	创建者，System代表系统自带。

状态码： 400

表 9-14 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-15 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

状态码: 200

Success

```
{
  "count" : 2,
  "resources" : [ {
    "id" : 1012384820829081600,
    "name" : "test",
    "dimension" : "Uniqueness",
    "sql_info" : "select count(*) from ${Schema_Table1}",
    "category_id" : 0,
    "type" : "Table",
    "system_template" : false,
    "create_time" : 1661412950486,
    "creator" : "ei",
    "result_description" : "row : ${1}"
  }, {
    "id" : 1,
    "name" : "表行数",
    "dimension" : "Accuracy",
    "sql_info" : "select count(1) from ${Schema_Table1}",
    "category_id" : 0,
    "type" : "Table",
    "system_template" : true,
    "create_time" : 1571328000000,
    "creator" : "System",
    "result_description" : "总行数 : ${1}"
  }
]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListQualityTemplatesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListQualityTemplatesRequest request = new ListQualityTemplatesRequest();
try {
    ListQualityTemplatesResponse response = client.listQualityTemplates(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListQualityTemplatesRequest()
        response = client.list_quality_templates(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListQualityTemplatesRequest{}
    response, err := client.ListQualityTemplates(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.2.2 创建规则模板

### 功能介绍

创建规则模板。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/quality/rule-templates

表 9-16 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 9-17 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-18 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	规则模板ID。
name	否	String	规则模板名称。
directory_id	否	Long	目录。
dimension_id	否	Integer	维度ID，1表示完整性，2表示唯一性，3表示及时性，4表示有效性，5表示准确性，6表示一致性。
description	否	String	描述。
sql_info	否	String	模板中的sql内容。
result_description	否	String	结果说明。

参数	是否必选	参数类型	描述
publish	否	Boolean	是否是发布操作，true表示发布新版本，false表示普通的保存操作。
origin_name	否	String	修改前的模板名。
abnormal_table_template	否	String	异常表模板。
user_define_version_name	否	String	用户自定义版本名。
version_num	否	Long	获取模板信息时候的版本号。
status	否	Integer	规则模板状态，0表示下线，1表示已发布。

## 响应参数

状态码： 200

表 9-19 响应 Body 参数

参数	参数类型	描述
id	Long	规则模板ID。
name	String	规则模板名称。
category_id	Long	目录ID
dimension	String	维度，Completeness表示完整性，Uniqueness表示唯一性，Timeliness表示及时性，Validity表示有效性，Accuracy表示准确性，Consistency表示一致性。
type	String	规则类型，Field表示字段级规则，Table表示表级规则，Database表示库级规则，Cross-field表示跨字段级规则，Customize表示自定义规则。
system_template	Boolean	是否为系统模板。
sql_info	String	定义关系。
abnormal_table_template	String	异常表模板。
result_description	String	结果说明。
create_time	Long	创建时间，13位时间戳（精确到毫秒）。



参数	参数类型	描述
creator	String	创建者，System代表系统自带。

状态码： 400

表 9-20 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-21 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

创建一个规则模板名称为test的规则模板，模板id为0，目录id为0，维度id为1，模板中的sql内容为select count(\*) from \${Schema\_Table1}，发布操作默认为false，结果说明为总条数，描述为test。

POST /v2/0833a5737480d53b2f25c010dc1a7b88/quality/rule-templates

```
{
  "id": "0",
  "name": "test",
  "directory_id": "0",
  "dimension_id": 1,
  "description": "test",
  "sql_info": "select count(*) from ${Schema_Table1}",
  "result_description": "总条数",
  "publish": false,
  "origin_name": null,
  "abnormal_table_template": "select count(*) from ${Schema_Table1}",
  "user_define_version_name": null,
  "version_num": null,
  "status": null
}
```

## 响应示例

状态码： 200

## Success

```
{
  "id": "1045431715138617345",
  "name": "test",
  "category_id": 0,
  "dimension": "Completeness",
  "type": "Table",
  "system_template": false,
  "sql_info": "select count(*) from ${Schema_Table1}",
  "abnormal_table_template": "select count(*) from ${Schema_Table1}",
  "result_description": "总条数",
  "create_time": 1640678720014,
  "creator": "test"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

创建一个规则模板名称为test的规则模板，模板id为0，目录id为0，维度id为1，模板中的sql内容为select count(\*) from \${Schema\_Table1}，发布操作默认为false，结果说明为总条数，描述为test。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))
            .build();
        CreateTemplateRequest request = new CreateTemplateRequest();
        TemplateRO body = new TemplateRO();
        body.withAbnormalTableTemplate("select count(*) from ${Schema_Table1}");
        body.withPublish(false);
        body.withResultDescription("总条数");
        body.withSqlInfo("select count(*) from ${Schema_Table1}");
        body.withDescription("test");
        body.withDimensionId(1);
    }
}
```

```
body.withDirectoryId(0L);
body.withName("test");
body.withId("0");
request.withBody(body);
try {
    CreateTemplateResponse response = client.createTemplate(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

创建一个规则模板名称为test的规则模板，模板id为0，目录id为0，维度id为1，模板中的sql内容为select count(\*) from \${Schema\_Table1}，发布操作默认为false，结果说明为总条数，描述为test。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateTemplateRequest()
        request.body = TemplateRO(
            abnormal_table_template="select count(*) from ${Schema_Table1}",
            publish=False,
            result_description="总条数",
            sql_info="select count(*) from ${Schema_Table1}",
            description="test",
            dimension_id=1,
            directory_id=0,
            name="test",
            id="0"
        )
        response = client.create_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

创建一个规则模板名称为test的规则模板，模板id为0，目录id为0，维度id为1，模板中的sql内容为select count(\*) from \${Schema\_Table1}，发布操作默认为false，结果说明为总条数，描述为test。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateTemplateRequest{}
    abnormalTableTemplateTemplateRo:= "select count(*) from ${Schema_Table1}"
    publishTemplateRo:= false
    resultDescriptionTemplateRo:= "总条数"
    sqlInfoTemplateRo:= "select count(*) from ${Schema_Table1}"
    descriptionTemplateRo:= "test"
    dimensionIdTemplateRo:= int32(1)
    directoryIdTemplateRo:= int64(0)
    nameTemplateRo:= "test"
    idTemplateRo:= "0"
    request.Body = &model.TemplateRo{
        AbnormalTableTemplate: &abnormalTableTemplateTemplateRo,
        Publish: &publishTemplateRo,
        ResultDescription: &resultDescriptionTemplateRo,
        SqlInfo: &sqlInfoTemplateRo,
        Description: &descriptionTemplateRo,
        DimensionId: &dimensionIdTemplateRo,
        DirectoryId: &directoryIdTemplateRo,
        Name: &nameTemplateRo,
        Id: &idTemplateRo,
    }
    response, err := client.CreateTemplate(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
```

```
    fmt.Println(err)
  }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.2.3 获取规则模板详情

### 功能介绍

获取规则模板详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/rule-templates/{id}

表 9-22 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	Long	规则模板ID。

## 请求参数

表 9-23 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-24 响应 Body 参数

参数	参数类型	描述
id	Integer	规则模板ID。
name	String	规则模板名称。
category_id	Long	目录ID。
dimension	String	维度，Completeness表示完整性，Uniqueness表示唯一性，Timeliness表示及时性，Validity表示有效性，Accuracy表示准确性，Consistency表示一致性。
type	String	规则类型，Field表示字段级规则，Table表示表级规则，Database表示库级规则，Cross-field表示跨字段级规则，Customize表示自定义规则。
system_template	Boolean	是否为系统模板。
sql_info	String	定义关系。
result_description	String	结果说明。
create_time	Long	创建时间，13位时间戳（精确到毫秒）。
creator	String	创建者，System代表系统自带。

状态码： 400

表 9-25 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-26 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

状态码： 200

Success

```
{
  "id" : 1012384820829081600,
  "name" : "test",
  "dimension" : "Uniqueness",
  "sql_info" : "select count(*) from ${Schema_Table1}",
  "category_id" : 0,
  "type" : "Table",
  "system_template" : false,
  "create_time" : 1661412950486,
  "creator" : "ei",
  "result_description" : "row : ${1}"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowTemplatesDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTemplatesDetailRequest request = new ShowTemplatesDetailRequest();
        request.withId({id}L);
        try {
            ShowTemplatesDetailResponse response = client.showTemplatesDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
```



```
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ShowTemplatesDetailRequest()
    request.id = {id}
    response = client.show_templates_detail(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowTemplatesDetailRequest{}
    request.Id = int64({id})
    response, err := client.ShowTemplatesDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.2.4 更新规则模板

### 功能介绍

更新规则模板

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/quality/rule-templates/{id}

表 9-27 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	Long	规则模板ID。

### 请求参数

表 9-28 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-29 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	规则模板ID。
name	否	String	规则模板名称。
directory_id	否	Long	目录。
dimension_id	否	Integer	维度ID, 1表示完整性, 2表示唯一性, 3表示及时性, 4表示有效性, 5表示准确性, 6表示一致性。
description	否	String	描述。
sql_info	否	String	模板中的sql内容。
result_description	否	String	结果说明。
publish	否	Boolean	是否是发布操作, true表示发布新版本, false表示普通的保存操作。
origin_name	否	String	修改前的模板名。
abnormal_table_template	否	String	异常表模板。
user_define_version_name	否	String	用户自定义版本名。
version_num	否	Long	获取模板信息时候的版本号。
status	否	Integer	规则模板状态, 0表示下线, 1表示已发布。

## 响应参数

状态码： 200

表 9-30 响应 Body 参数

参数	参数类型	描述
id	Long	规则模板ID。
name	String	规则模板名称。
category_id	Long	目录ID

参数	参数类型	描述
dimension	String	维度，Completeness表示完整性，Uniqueness表示唯一性，Timeliness表示及时性，Validity表示有效性，Accuracy表示准确性，Consistency表示一致性。
type	String	规则类型，Field表示字段级规则，Table表示表级规则，Database表示库级规则，Cross-field表示跨字段级规则，Customize表示自定义规则。
system_template	Boolean	是否为系统模板。
sql_info	String	定义关系。
abnormal_table_template	String	异常表模板。
result_description	String	结果说明。
create_time	Long	创建时间，13位时间戳（精确到毫秒）。
creator	String	创建者，System代表系统自带。

状态码： 400

表 9-31 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-32 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

更新规则模板名称为test的规则模板，模板id为0，目录id为0，维度id为1，模板中的sql内容为select count(\*) from \${Schema\_Table1}，发布操作默认为false，结果说明为总条数，描述为test。

```
PUT /v2/0833a5737480d53b2f25c010dc1a7b88/quality/rule-templates/1045431715138617345
{
  "id": "1045431715138617345",
  "name": "test",
  "directory_id": "0",
  "dimension_id": 1,
  "description": "test",
  "sql_info": "select count(*) from ${Schema_Table1}",
  "result_description": "总条数",
  "publish": true,
  "origin_name": "test",
  "abnormal_table_template": "select count(*) from ${Schema_Table1}",
  "user_define_version_name": "V1.1",
  "version_num": 0,
  "status": 1
}
```

## 响应示例

状态码： 200

Success

```
{
  "id": "1045431715138617345",
  "name": "test",
  "category_id": 0,
  "dimension": "Completeness",
  "type": "Table",
  "system_template": false,
  "sql_info": "select count(*) from ${Schema_Table1}",
  "abnormal_table_template": "select count(*) from ${Schema_Table1}",
  "result_description": "总条数",
  "create_time": 1640678720014,
  "creator": "test"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新规则模板名称为test的规则模板，模板id为0，目录id为0，维度id为1，模板中的sql内容为select count(\*) from \${Schema\_Table1}，发布操作默认为false，结果说明为总条数，描述为test。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class UpdateTemplateSolution {
    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))
            .build();
        UpdateTemplateRequest request = new UpdateTemplateRequest();
        request.withId("{id}");
        TemplateRO body = new TemplateRO();
        body.withStatus(1);
        body.withVersionNum(0L);
        body.withUserDefineVersionName("V1.1");
        body.withAbnormalTableTemplate("select count(*) from ${Schema_Table1}");
        body.withOriginName("test");
        body.withPublish(true);
        body.withResultDescription("总条数");
        body.withSqlInfo("select count(*) from ${Schema_Table1}");
        body.withDescription("test");
        body.withDimensionId(1);
        body.withDirectoryId(0L);
        body.withName("test");
        body.withId("1045431715138617345");
        request.withBody(body);
        try {
            UpdateTemplateResponse response = client.updateTemplate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

更新规则模板名称为test的规则模板，模板id为0，目录id为0，维度id为1，模板中的sql内容为select count(\*) from \${Schema\_Table1}，发布操作默认为false，结果说明为总条数，描述为test。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateTemplateRequest()
        request.id = "{id}"
        request.body = TemplateRO(
            status=1,
            version_num=0,
            user_define_version_name="V1.1",
            abnormal_table_template="select count(*) from ${Schema_Table1}",
            origin_name="test",
            publish=True,
            result_description="总条数",
            sql_info="select count(*) from ${Schema_Table1}",
            description="test",
            dimension_id=1,
            directory_id=0,
            name="test",
            id="1045431715138617345"
        )
        response = client.update_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

更新规则模板名称为test的规则模板，模板id为0，目录id为0，维度id为1，模板中的sql内容为select count(\*) from \${Schema\_Table1}，发布操作默认为false，结果说明为总条数，描述为test。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```

ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateTemplateRequest{
    request.Id = "{id}"
    statusTemplateRo:= int32(1)
    versionNumTemplateRo:= int64(0)
    userDefineVersionNameTemplateRo:= "V1.1"
    abnormalTableTemplateTemplateRo:= "select count(*) from ${Schema_Table1}"
    originNameTemplateRo:= "test"
    publishTemplateRo:= true
    resultDescriptionTemplateRo:= "总条数"
    sqlInfoTemplateRo:= "select count(*) from ${Schema_Table1}"
    descriptionTemplateRo:= "test"
    dimensionIdTemplateRo:= int32(1)
    directoryIdTemplateRo:= int64(0)
    nameTemplateRo:= "test"
    idTemplateRo:= "1045431715138617345"
    request.Body = &model.TemplateRo{
        Status: &statusTemplateRo,
        VersionNum: &versionNumTemplateRo,
        UserDefineVersionName: &userDefineVersionNameTemplateRo,
        AbnormalTableTemplate: &abnormalTableTemplateTemplateRo,
        OriginName: &originNameTemplateRo,
        Publish: &publishTemplateRo,
        ResultDescription: &resultDescriptionTemplateRo,
        SqlInfo: &sqlInfoTemplateRo,
        Description: &descriptionTemplateRo,
        DimensionId: &dimensionIdTemplateRo,
        DirectoryId: &directoryIdTemplateRo,
        Name: &nameTemplateRo,
        Id: &idTemplateRo,
    }
}
response, err := client.UpdateTemplate(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest



状态码	描述
500	INTERNAL SERVER ERROR

## 9.2.5 批量删除规则模板

### 功能介绍

批量删除规则模板。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/quality/rule-templates/batch-delete

表 9-33 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 9-34 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-35 请求 Body 参数

参数	是否必选	参数类型	描述
ids	否	Array of longs	模板ID列表。

## 响应参数

状态码： 400

表 9-36 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-37 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

批量删除规则模板，模板id列表为[1045431715138617345, 1035847774756868096]。

```
POST /v2/0833a5737480d53b2f25c010dc1a7b88/quality/rule-templates/batch-delete
{
  "ids" : [ 1045431715138617345, 1035847774756868096 ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

批量删除规则模板，模板id列表为[1045431715138617345, 1035847774756868096]。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteTemplatesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteTemplatesRequest request = new BatchDeleteTemplatesRequest();
        TemplateListRO body = new TemplateListRO();
        List<Long> listbodyIds = new ArrayList<>();
        listbodyIds.add(1045431715138617345L);
        listbodyIds.add(1035847774756868096L);
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            BatchDeleteTemplatesResponse response = client.batchDeleteTemplates(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

批量删除规则模板，模板id列表为[1045431715138617345, 1035847774756868096]。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = BatchDeleteTemplatesRequest()
    listIdsbody = [
        1045431715138617345,
        1035847774756868096
    ]
    request.body = TemplateListRO(
        ids=listIdsbody
    )
    response = client.batch_delete_templates(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

批量删除规则模板，模板id列表为[1045431715138617345, 1035847774756868096]。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteTemplatesRequest{}
```

```
var listIdsbody = []int64{
    int64(1045431715138617345),
    int64(1035847774756868096),
}
request.Body = &model.TemplateListRo{
    Ids: &listIdsbody,
}
response, err := client.BatchDeleteTemplates(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.3 质量作业接口

### 9.3.1 获取质量作业列表 V1

#### 功能介绍

批量查询数据质量作业，返回质量作业列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/quality/rules

表 9-38 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-39 Query 参数

参数	是否必选	参数类型	描述
start	否	Long	开始查询页数，用于计算分页查询偏移量，如果参数offset和limit均不为空则以offset值为准。
page_size	否	Long	每页显示条目数量，如果参数offset和limit均不为空则以limit值为准。
offset	否	Integer	分页查询偏移量。
limit	否	Integer	每页显示的条目数量。
category_id	否	Long	目录ID。
rule_name	否	String	质量作业名称。
schedule_status	否	Integer	调度状态，1表示未启动，2表示调度中，3表示正常结束，4表示手动停止，5表示调度异常。
schedule_period	否	Integer	调度周期，1表示分钟，2表示小时，3表示天，4表示周，5表示月。
start_time	否	String	开始时间（搜索），格式 yyyy-MM-dd HH:mm:ss。
end_time	否	String	结束时间（搜索），格式 yyyy-MM-dd HH:mm:ss。
result_status	否	Integer	最近运行结果，0表示运行中，1表示异常，2表示告警，3表示正常。
sort	否	String	排序字段。
order	否	String	排序方式。

### 请求参数

无

### 响应参数

状态码： 200

表 9-40 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码： 400

表 9-41 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码： 401

表 9-42 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码： 403

表 9-43 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。
data	Object	返回的数据信息。

## 请求示例

无

## 响应示例

状态码： 200

Success, 返回的data数据。

```
{
  "data" : {
    "num" : 100,
    "page_search_result" : [ {
      "id" : "1042745498940891137",
      "name" : "rule-name",
      "creator" : "test-user",
      "create_offset_time" : "2022-11-17T10:23:12+08:00",
      "modify_offset_time" : "2022-11-17T10:23:46+08:00",
      "dw_name" : "",
      "db_name" : "",
      "schedule_status" : 1,
      "calc_offset_time" : "2022-11-17T10:51:59+08:00"
    } ]
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListQualityTaskListsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
    }
}
```



```
ListQualityTaskListsRequest request = new ListQualityTaskListsRequest();
try {
    ListQualityTaskListsResponse response = client.listQualityTaskLists(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListQualityTaskListsRequest()
        response = client.list_quality_task_lists(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListQualityTaskListsRequest{}
response, err := client.ListQualityTaskLists(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success, 返回的data数据。
400	BadRequest
401	Unauthorized
403	Forbidden

### 9.3.2 获取质量作业列表

#### 功能介绍

获取质量作业列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/quality/quality-tasks

表 9-44 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-45 Query 参数

参数	是否必选	参数类型	描述
category_id	否	Long	目录ID。
name	否	String	质量作业名称。
schedule_status	否	String	调度状态：UNKNOWN表示未知，NOT_START表示未启动，SCHEDULING表示调度中，FINISH_SUCCESS表示正常结束，KILL表示手动停止，RUNNING_EXCEPTION表示运行失败。
start_time	否	Long	最近运行时间查询区间的开始时间，13位时间戳(精确到毫秒)。
end_time	否	Long	最近运行时间查询区间的结束时间，13位时间戳(精确到毫秒)。
creator	否	String	创建人
limit	否	Long	分页条数，最大值为100。
offset	否	Long	分页偏移量，最小值0。

## 请求参数

表 9-46 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-47 响应 Body 参数

参数	参数类型	描述
count	Long	总条数。
resources	Array of <a href="#">QualityTaskOverviewVO</a> objects	分页数据。

表 9-48 QualityTaskOverviewVO

参数	参数类型	描述
id	Long	质量作业ID。
name	String	质量作业名称。
category_id	Long	目录ID。
schedule_status	String	调度状态，UNKNOWN表示未知，NOT_START表示未启动，SCHEDULING表示调度中，FINISH_SUCCESS表示正常结束，KILL表示手动停止，RUNNING_EXCEPTION表示运行失败。
schedule_period	String	调度周期，MINUTE表示按分钟调度，HOUR表示按小时调度，DAY表示按天调度，WEEK表示按周调度。
schedule_interval	String	调度间隔，当调度周期为分钟、小时、天时，间隔时间为数字，而当调度周期为周时，调度间隔为星期的英文，如：每周一、周二调度时，schedule_interval为"MONDAY,TUESDAY"。
create_time	Long	创建时间，13位时间戳（精确到毫秒）。
last_run_time	Long	最近运行时间，13位时间戳（精确到毫秒）。
creator	String	创建者。

状态码： 400

表 9-49 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。

参数	参数类型	描述
error_msg	String	错误信息。

状态码： 500

表 9-50 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

状态码： 200

Success 返回的是 InstanceVO。

```
{
  "count" : 1,
  "resources" : [ {
    "id" : 1012385458115825700,
    "name" : "test",
    "category_id" : 1012384968858652700,
    "schedule_status" : "NOT_START",
    "schedule_period" : null,
    "schedule_interval" : null,
    "create_time" : 1661413102000,
    "last_run_time" : 1661413281000,
    "creator" : "ei"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class ListQualityTaskSolution {
    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListQualityTaskRequest request = new ListQualityTaskRequest();
        try {
            ListQualityTaskResponse response = client.listQualityTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = ListQualityTaskRequest()
response = client.list_quality_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListQualityTaskRequest{}
    response, err := client.ListQualityTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success 返回的是 InstanceVO。
400	BadRequest

状态码	描述
500	INTERNAL SERVER ERROR

### 9.3.3 获取质量作业详情

#### 功能介绍

获取质量作业详情

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/quality/quality-tasks/{id}

表 9-51 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	质量作业ID。

#### 请求参数

表 9-52 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

#### 响应参数

状态码： 200



表 9-53 响应 Body 参数

参数	参数类型	描述
id	Long	作业ID。
name	String	作业名称。
description	String	作业描述。
category_id	Long	所属目录ID。
level	String	告警级别，SUGGEST表示提示，MINOR表示一般，MAJOR表示严重，FATAL表示致命。
alarm_condition	String	统一告警条件。
alarm_notify	Boolean	是否开启通知告警。
alarm_notify_type	String	告警通知类型，TRIGGER_ALARM表示触发告警，RUN_SUCCESS表示运行成功，TRIGGER_ALARM_AND_RUNNING_SUCCESS表示触发告警和运行成功。
alarm_notify_topic	String	通知主题名。
schedule_type	String	调度类型，ONCE表示单次调度，PERIODIC表示周期性调度。
schedule_start_date	String	调度开始日期。
schedule_end_date	String	调度结束日期。
schedule_period	String	调度周期，MINUTE表示按分钟调度，HOUR表示按小时调度，DAY表示按天调度，WEEK表示按周调度。
schedule_interval	String	调度间隔，当调度周期为分钟、小时、天时，间隔时间为数字，而当调度周期为周时，调度间隔为星期的英文，如：每周一、周二调度时，schedule_interval为"MONDAY,TUESDAY"。
schedule_start_time	String	调度开始时间。
schedule_end_time	String	调度结束时间。
create_time	Long	最近运行时间，13位时间戳（精确到毫秒）。
last_run_time	Long	最近运行时间，13位时间戳（精确到毫秒）。

参数	参数类型	描述
sub_rules	Array of <a href="#">QualityTaskRuleDetailForOpenApi</a> objects	子规则。
schedule_cron	String	调度cron表达式。

表 9-54 QualityTaskRuleDetailForOpenApi

参数	参数类型	描述
id	Long	子规则ID。
sub_rule_name	String	子规则名称。
type	String	规则类型，SingleDatabase表示库级规则，SingleTable表示表级规则，SingleColumn表示字段级规则，CrossColumn表示跨字段规则，Customize表示自定义规则。
template_id	Long	规则模板ID。
connection	String	数据连接ID。
connection_type	String	数据连接类型。
databases	String	数据库名，当存在多个值时以逗号分隔。
sql	String	自定义规则中的SQL脚本，系统内置规则时返回。
tables	String	表名，当存在多个值时以逗号分隔。
reference_tables	String	跨字段规则中的参考表名，当存在多个值时以逗号分隔。
columns	String	字段名，当存在多个值时以逗号分隔。
reference_columns	String	跨字段规则中的参考字段名，当存在多个值时以逗号分隔。
ignore_error	Boolean	是否忽视规则错误。
dimension	String	维度，Completeness表示完整性，Uniqueness表示唯一性，Timeliness表示及时性，Validity表示有效性，Accuracy表示准确性，Consistency表示一致性。
queue	String	DLI队列。

参数	参数类型	描述
regular_expression	String	当规则模板为正则表达式校验时的正则表达式。
template_arguments	String	模板参数。
weight	Integer	规则权重。
calculation_range	String	计算范围。
calculation_range_sql	String	计算范围SQL。
alarm_condition	String	告警表达式。
export_abnormal_table	Boolean	是否导出异常数据。
abnormal_table_database	String	异常表数据库。
abnormal_table_schema	String	异常表Schema。
abnormal_table	String	异常字段所在的表。
abnormal_table_prefix	String	异常表前缀。
abnormal_table_suffix	String	异常表后缀。
abnormal_table_columns	String	异常字段名，当存在多个值时以逗号分隔。
abnormal_table_sql	String	异常表SQL。
abnormal_table_output_config	Boolean	异常表是否输出规则配置。
abnormal_table_include_null_value	Boolean	异常表是否包含空值。
abnormal_table_output_data_number	Integer	异常表输出行数，0代表全量输出。
score_switch	Boolean	是否开启质量评分。
score_schema	String	质量评分表所在schema。
score_table	String	质量评分表名。
score_expression	String	质量评分表达式。

状态码： 400

表 9-55 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-56 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

状态码： 200

Success 返回的是 InstanceVO。

```
{
  "id": 1012385458115825700,
  "name": "test",
  "description": "",
  "level": "SUGGEST",
  "alarm_condition": "",
  "alarm_notify": false,
  "alarm_notify_type": null,
  "alarm_notify_topic": "",
  "schedule_type": "ONCE",
  "schedule_start_date": null,
  "schedule_end_date": null,
  "schedule_period": null,
  "schedule_interval": null,
  "schedule_start_time": null,
  "schedule_end_time": null,
  "category_id": 1012384968858652700,
  "create_time": 1661413102000,
  "last_run_time": 1661413281000,
  "sub_rules": [ {
    "id": 1012385458417815600,
    "sub_rule_name": "qwe",
    "type": "SingleTable",
    "connection": "43b9e6315eda4918970cfcbbfb74cb1",
    "connection_type": "DWS",
    "databases": "1_database",
    "sql": "select count(1) from 1_database.dqc_test2 where 1=1",
    "tables": "1_database.dqc_test2",
```

```
"reference_tables" : null,
"columns" : "",
"reference_columns" : null,
"ignore_error" : false,
"dimension" : "Accuracy",
"queue" : "dlf_notdelete",
"regular_expression" : null,
"template_arguments" : "",
"weight" : 5,
"calculation_range" : "FULL_SCAN",
"calculation_range_sql" : null,
"alarm_condition" : "${1}<0",
"export_abnormal_table" : false,
"abnormal_table_database" : null,
"abnormal_table_schema" : null,
"abnormal_table" : "",
"abnormal_table_prefix" : null,
"abnormal_table_suffix" : null,
"abnormal_table_columns" : null,
"abnormal_table_sql" : "",
"abnormal_table_out_config" : false,
"abnormal_table_include_null_value" : false,
"abnormal_table_out_data_number" : 0,
"score_switch" : false,
"score_schema" : null,
"score_table" : null,
"score_expression" : "",
"template_id" : 1
}],
"schedule_cron" : ""
}
```

## 状态码

状态码	描述
200	Success 返回的是 InstanceVO。
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.3.4 批量删除质量作业

### 功能介绍

批量删除质量作业

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/quality/quality-tasks/batch-delete

表 9-57 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 9-58 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-59 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of integers	批量删除的质量作业id

## 响应参数

状态码： 200

表 9-60 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 400

表 9-61 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-62 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

删除ID为1283480044387540993、1279033542491521025的质量作业。

```
POST https://{endpoint}/v2/{project_id}/quality/quality-tasks/batch-delete
[ 1283480044387540993, 1279033542491521025 ]
```

## 响应示例

状态码： 200

Success

```
{
  "error_code" : "DLG.2700",
  "error_msg" : "成功"
}
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.4 对账作业接口

## 9.4.1 获取对账作业列表

### 功能介绍

获取对账作业列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/consistency-tasks

表 9-63 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-64 Query 参数

参数	是否必选	参数类型	描述
category_id	否	Long	目录ID。
name	否	String	name。
schedule_status	否	String	调度状态：UNKNOWN表示未知，NOT_START表示未启动，SCHEDULING表示调度中，FINISH_SUCCESS表示正常结束，KILL表示手动停止，RUNNING_EXCEPTION表示运行失败。
start_time	否	Long	最近运行时间查询区间的开始时间，13位时间戳(精确到毫秒)。
end_time	否	Long	最近运行时间查询区间的结束时间，13位时间戳(精确到毫秒)。
creator	否	String	创建人。
limit	否	Long	分页条数，最大值为100。
offset	否	Long	分页偏移量，最小值0



## 请求参数

表 9-65 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-66 响应 Body 参数

参数	参数类型	描述
count	Long	总条数。
resources	Array of <a href="#">QualityTaskOverviewVO_2</a> objects	分页数据。

表 9-67 QualityTaskOverviewVO\_2

参数	参数类型	描述
id	Long	对账作业ID。
name	String	对账作业名称。
category_id	Long	目录ID。
schedule_status	String	调度状态，UNKNOWN表示未知，NOT_START表示未启动，SCHEDULING表示调度中，FINISH_SUCCESS表示正常结束，KILL表示手动停止，RUNNING_EXCEPTION表示运行失败。
schedule_period	String	调度周期，MINUTE表示按分钟调度，HOUR表示按小时调度，DAY表示按天调度，WEEK表示按周调度。

参数	参数类型	描述
schedule_interval	String	调度间隔，当调度周期为分钟、小时、天时，间隔时间为数字，而当调度周期为周时，调度间隔为星期的英文，如：每周一、周二调度时，schedule_interval 为"MONDAY,TUESDAY"。
create_time	Long	创建时间，13位时间戳（精确到毫秒）。
last_run_time	Long	最新运行时间，13位时间戳（精确到毫秒）。
creator	String	创建者。

状态码： 400

表 9-68 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-69 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

状态码： 200

Success 返回的是InstanceVO

```
{
  "count" : 1,
```

```
"resources" : [ {  
  "id" : 1012386182149165000,  
  "name" : "test",  
  "category_id" : 0,  
  "schedule_status" : "NOT_START",  
  "schedule_period" : null,  
  "schedule_interval" : null,  
  "create_time" : 1661413275050,  
  "last_run_time" : null,  
  "creator" : "ei"  
} ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListConsistencyTaskSolution {  
  
  public static void main(String[] args) {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    // environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running  
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    String ak = System.getenv("CLOUD_SDK_AK");  
    String sk = System.getenv("CLOUD_SDK_SK");  
    String projectId = "{project_id}";  
  
    ICredential auth = new BasicCredentials()  
      .withProjectId(projectId)  
      .withAk(ak)  
      .withSk(sk);  
  
    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
      .withCredential(auth)  
      .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
      .build();  
    ListConsistencyTaskRequest request = new ListConsistencyTaskRequest();  
    try {  
      ListConsistencyTaskResponse response = client.listConsistencyTask(request);  
      System.out.println(response.toString());  
    } catch (ConnectionException e) {  
      e.printStackTrace();  
    } catch (RequestTimeoutException e) {  
      e.printStackTrace();  
    } catch (ServiceResponseException e) {  
      e.printStackTrace();  
      System.out.println(e.getStatusCode());  
      System.out.println(e.getRequestId());  
      System.out.println(e.getErrorCode());  
      System.out.println(e.getErrorMsg());  
    }  
  }  
}
```

```
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListConsistencyTaskRequest()  
        response = client.list_consistency_task(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dataartsstudio.NewDataArtsStudioClient(  

```

```
dataartsstudio.DataArtsStudioClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build()

request := &model.ListConsistencyTaskRequest{}
response, err := client.ListConsistencyTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success 返回的是InstanceVO
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.4.2 获取对账作业详情

### 功能介绍

获取对账作业详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/consistency-tasks/{id}

表 9-70 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
id	是	String	对账作业ID。

## 请求参数

表 9-71 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-72 响应 Body 参数

参数	参数类型	描述
id	Long	作业ID。
name	String	作业名称。
description	String	作业描述。
category_id	Long	所属目录ID
level	String	告警级别，SUGGEST表示提示，MINOR表示一般，MAJOR表示严重，FATAL表示致命。
alarm_condition	String	统一告警条件。
alarm_notify	Boolean	是否开启通知告警。
alarm_notify_type	String	告警通知类型，TRIGGER_ALARM表示触发告警，RUN_SUCCESS表示运行成功，TRIGGER_ALARM_AND_RUNNING_SUCCESS表示触发告警和运行成功。
alarm_notify_topic	String	通知主题名
schedule_type	String	调度类型，ONCE表示单次调度，PERIODIC表示周期性调度。
schedule_period	String	调度周期，MINUTE表示按分钟调度，HOUR表示按小时调度，DAY表示按天调度，WEEK表示按周调度。

参数	参数类型	描述
schedule_interval	String	调度间隔，注意：当调度周期为分钟、小时、天时，间隔时间为数字；而当调度周期为周时，调度间隔为星期的英文，如：每周一、周二调度时，schedule_interval为"MONDAY,TUESDAY"。
schedule_start_time	String	调度开始时间。
schedule_end_time	String	调度结束时间。
create_time	Long	最近运行时间，13位时间戳（精确到毫秒）。
last_run_time	Long	最近运行时间，13位时间戳（精确到毫秒）。
sub_rules	Array of <a href="#">ConsistencyRuleDetailForOpenApi</a> objects	子规则。

表 9-73 ConsistencyRuleDetailForOpenApi

参数	参数类型	描述
id	Long	子规则ID。
sub_rule_name	String	子规则名称。
type	String	规则类型，SingleDatabase表示库级规则，SingleTable表示表级规则，SingleColumn表示字段级规则，CrossColumn表示跨字段规则，Customize表示自定义规则。
template_id	Long	规则模板ID。
connection	String	数据连接ID。
connection_type	String	数据连接类型。
databases	String	数据库名，当存在多个值时以逗号分隔。
sql	String	自定义规则中的SQL脚本。
tables	String	表名，当存在多个值时以逗号分隔。
reference_tables	String	对照表名，当存在多个值时以逗号分隔。

参数	参数类型	描述
columns	String	字段名，当存在多个值时以逗号分隔。
reference_columns	String	对照列名，当存在多个值时以逗号分隔。
dimension	String	维度。
queue	String	DLI队列。
regular_expression	String	当规则模板为正则表达式校验时的正则表达式。
template_arguments	String	模板参数。
weight	Integer	规则权重。
calculation_range	String	计算范围。
calculation_range_sql	String	计算范围SQL。
alarm_condition	String	告警表达式。

状态码： 400

表 9-74 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-75 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无



## 响应示例

**状态码： 200**

Success 返回的是 InstanceVO。

```
{
  "id" : 1012386182149165000,
  "name" : "test",
  "description" : "",
  "level" : "SUGGEST",
  "alarm_condition" : null,
  "alarm_notify" : false,
  "alarm_notify_type" : null,
  "alarm_notify_topic" : null,
  "schedule_type" : "ONCE",
  "schedule_period" : null,
  "schedule_interval" : null,
  "schedule_start_time" : null,
  "schedule_end_time" : null,
  "category_id" : 0,
  "create_time" : 1661413275050,
  "last_run_time" : null,
  "sub_rules" : [ {
    "id" : 1012386182602149900,
    "sub_rule_name" : "qew",
    "type" : "SingleTable",
    "connection" : "43b9e6315eda4918970cfcbbbf74cb1",
    "connection_type" : "DWS",
    "databases" : "1_database",
    "sql" : "",
    "tables" : "1_database.dqc_test2",
    "reference_tables" : null,
    "columns" : "",
    "reference_columns" : null,
    "dimension" : null,
    "queue" : "dlf_notdelete",
    "regular_expression" : null,
    "template_arguments" : null,
    "weight" : 5,
    "calculation_range" : "FULL_SCAN",
    "calculation_range_sql" : "",
    "alarm_condition" : "${1_1}>${2_1}",
    "template_id" : 1
  }, {
    "id" : 1012386182631510000,
    "sub_rule_name" : "qew",
    "type" : "SingleTable",
    "connection" : "43b9e6315eda4918970cfcbbbf74cb1",
    "connection_type" : "DWS",
    "databases" : "1_database",
    "sql" : "",
    "tables" : "1_database.dqc_test2",
    "reference_tables" : null,
    "columns" : "",
    "reference_columns" : null,
    "dimension" : null,
    "queue" : "dlf_notdelete",
    "regular_expression" : null,
    "template_arguments" : null,
    "weight" : 5,
    "calculation_range" : "FULL_SCAN",
    "calculation_range_sql" : "",
    "alarm_condition" : "${1_1}>${2_1}",
    "template_id" : 1
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowConsistencyTaskDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowConsistencyTaskDetailRequest request = new ShowConsistencyTaskDetailRequest();
        request.withId("{id}");
        try {
            ShowConsistencyTaskDetailResponse response = client.showConsistencyTaskDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowConsistencyTaskDetailRequest()
        request.id = "{id}"
        response = client.show_consistency_task_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowConsistencyTaskDetailRequest{}
    request.Id = "{id}"
    response, err := client.ShowConsistencyTaskDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success 返回的是 InstanceVO。
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.4.3 批量删除对账作业

### 功能介绍

批量删除对账作业

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/quality/consistency-tasks/batch-delete

表 9-76 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 9-77 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token, Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-78 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of integers	批量删除的对账作业id

## 响应参数

状态码： 200

表 9-79 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 400

表 9-80 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-81 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码, 如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

删除ID为1283480044387540993、1279033542491521025的对账作业。

```
POST https://{endpoint}/v2/{project_id}/quality/consistency-tasks/batch-delete
[ 1283480044387540993, 1279033542491521025 ]
```

## 响应示例

状态码: 200

Success

```
{
  "error_code": "DLG.2700",
  "error_msg": "成功"
}
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.5 运维管理接口

### 9.5.1 获取任务执行结果列表

#### 功能介绍

获取任务执行结果列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/quality/instances

表 9-82 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-83 Query 参数

参数	是否必选	参数类型	描述
name	否	String	规则名称。
task_type	否	String	任务实例类型，QUALITY_TASK 表示质量作业，CONSISTENCY_TASK 表示对账作业。
run_status	否	String	状态，RUNNING 表示运行中，FAILED 表示失败，ALARMING 表示报警，SUCCESS 表示正常，SUSPENDING 表示暂停中，UNKNOWN 表示未定义。
notify_status	否	String	通知状态，NOT_TRIGGERED 表示未触发，SUCCESS 表示成功，FAILED 表示失败。
start_time	否	Long	最近运行时间查询区间的开始时间，13 位时间戳(精确到毫秒)。
end_time	否	Long	最近运行时间查询区间的结束时间，13 位时间戳(精确到毫秒)。
limit	否	Integer	每页显示的条目数量，最大值为 100。
offset	否	Integer	分页偏移量。

## 请求参数

表 9-84 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio 工作空间 ID，获取方法请参见 <a href="#">实例 ID 和工作空间 ID</a> 。
X-Auth-Token	是	String	IAM Token，Token 获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-85 响应 Body 参数

参数	参数类型	描述
count	Integer	个数。
resources	Array of <a href="#">InstanceOverviewVo</a> objects	resources。

表 9-86 InstanceOverviewVo

参数	参数类型	描述
id	Long	作业实例ID。
name	String	作业实例名称。
task_id	Long	task id
task_type	String	作业类型，QUALITY_TASK表示质量作业，CONSISTENCY_TASK表示对账作业。
run_status	String	作业运行状态，RUNNING表示运行中，FAILED表示失败，ALARMING表示报警，SUCCESS表示正常。
notify_status	String	提示状态，NOT_TRIGGERED表示未触发，SUCCESS表示成功，FAILED表示失败。
start_time	Long	开始时间。
end_time	Long	结束时间。

状态码： 400

表 9-87 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500



表 9-88 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

状态码： 200

Success

```
{
  "count" : 2,
  "resources" : [ {
    "id" : 1012386207306600400,
    "name" : "test-1",
    "task_id" : 1012385458115825700,
    "task_type" : "QUALITY_TASK",
    "run_status" : "SUCCESS",
    "notify_status" : "NOT_TRIGGERED",
    "start_time" : 1661413281000,
    "end_time" : 1661413381000
  }, {
    "id" : 1012386190504218600,
    "name" : "test-0",
    "task_id" : 1012386182149165000,
    "task_type" : "CONSISTENCY_TASK",
    "run_status" : "SUCCESS",
    "notify_status" : "NOT_TRIGGERED",
    "start_time" : 1661413277000,
    "end_time" : 1661413380000
  }
  ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListInstancesSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    ListInstancesRequest request = new ListInstancesRequest();
    try {
        ListInstancesResponse response = client.listInstances(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstancesRequest()
        response = client.list_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListInstancesRequest{}
    response, err := client.ListInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.5.2 获取实例结果

### 功能介绍

获取实例结果。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/instances/{instance\_id}/result

表 9-89 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	projectId。

### 请求参数

表 9-90 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

### 响应参数

状态码： 200

表 9-91 响应 Body 参数

参数	参数类型	描述
count	Integer	总数量。
resources	Array of <a href="#">SubInstanceResult</a> objects	子作业实例资源配置ID。

表 9-92 SubInstanceResult

参数	参数类型	描述
sub_rule_id	String	子规则ID。
sub_instance_instance_id	String	子规则实例ID。
abnormal_table_status	String	异常表任务状态，UNSUPPORTED表示不支持异常表，READY表示准备中，RUNNING表示运行中，FAILED表示失败，SUCCESS表示成功。
results	Array of objects	结果集。

状态码： 500

表 9-93 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

状态码： 200

Success

```
{
  "count" : 2,
  "resources" : [ {
    "sub_rule_id" : "1012386182602149888",
    "sub_instance_instance_id" : "1012386190718128129",
    "abnormal_table_status" : "UNSUPPORTED",
    "results" : [ {
      "Total Rows" : "4",
      "Status" : "false",
      "Name" : "1_database.dqc_test2"
    } ]
  }, {
    "sub_rule_id" : "1012386182631510017",
    "sub_instance_instance_id" : "1012386190718128128",
    "abnormal_table_status" : "UNSUPPORTED",
    "results" : [ {
      "Total Rows" : "4",
      "Status" : "false",
      "Name" : "1_database.dqc_test2"
    } ]
  } ]
}
```

```
    }  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowInstanceResultSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowInstanceResultRequest request = new ShowInstanceResultRequest();  
        request.withInstanceld("{instance_id}");  
        try {  
            ShowInstanceResultResponse response = client.showInstanceResult(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceResultRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_result(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceResultRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowInstanceResult(request)
```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success
500	INTERNAL SERVER ERROR

## 9.5.3 数据质量运维管理操作处理&记录

### 功能介绍

该接口用于数据质量监控实例的问题处理，在控制台上可通过“更多>处理&记录”进入问题处理界面。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/quality/instances/{instance\_id}

表 9-94 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。



## 请求参数

表 9-95 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-96 请求 Body 参数

参数	是否必选	参数类型	描述
detail	否	String	仅填写处理意见或者关闭问题时使用该字段，填写为相关意见或关闭说明。
operator	否	String	仅问题移交他人时使用该字段，选择移交负责人。
open	否	Boolean	选择问题处理方式。false表示关闭问题。true表示问题处理意见或移交他人。
cancel_run	否	Boolean	是否取消该任务，取消则立即停止运行该任务。false表示取消运行该任务，true表示不取消该任务。

## 响应参数

状态码： 200

表 9-97 响应 Body 参数

参数	参数类型	描述
data	String	返回信息。

状态码： 500

表 9-98 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

### 请求示例

无

### 响应示例

无

### 状态码

状态码	描述
200	Success
500	INTERNAL SERVER ERROR

## 9.6 任务实例接口

### 9.6.1 批量停止实例

#### 功能介绍

批量停止实例

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v2/{project\_id}/quality/instances/batch-stop

表 9-99 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 9-100 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-101 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of integers	批量停止的作业实例id

## 响应参数

状态码： 200

表 9-102 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。
data	Object	返回的数据信息。

状态码： 400

表 9-103 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码： 406**

**表 9-104** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**状态码： 500**

**表 9-105** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。
data	Object	返回的数据信息。

**请求示例**

停止ID为1317151269261262848、1317151258234322945的实例。

```
POST https://{endpoint}/v2/{project_id}/quality/instances/batch-stop
[ 1317151269261262848, 1317151258234322945 ]
```

**响应示例**

**状态码： 200**

Success 返回的是InstanceVO

```
{
  "error_code" : "DLG.2700",
  "error_msg" : "成功"
}
```

**状态码**

状态码	描述
200	Success 返回的是InstanceVO
400	BadRequest
406	NOT ACCEPTABLE

状态码	描述
500	INTERNAL SERVER ERROR

## 9.7 质量报告接口

### 9.7.1 获取质量报告评分体系

#### 功能介绍

获取质量报告评分体系。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/quality/report/scoring

表 9-106 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 9-107 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

#### 响应参数

状态码： 200

表 9-108 响应 Body 参数

参数	参数类型	描述
full_score	Integer	满分值。

状态码： 400

表 9-109 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-110 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

```
GET /v2/0833a5737480d53b2f25c010dc1a7b88/quality/report/scoring
```

## 响应示例

状态码： 200

Success

```
{  
  "full_score": 100  
}
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.7.2 获取质量报告概览

### 功能介绍

获取质量报告概览。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/report/overview

表 9-111 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-112 Query 参数

参数	是否必选	参数类型	描述
quality_score_dimension	是	String	质量评分维度。
l1	否	String	主题域分组(业务报告有效)。
l2	否	String	主题域(业务报告有效)。
l3	否	String	业务对象(业务报告有效)。
data_connection_id	否	String	数据连接id(技术报告有效)。
database_name	否	String	数据库名称(技术报告有效)。
table_name	否	String	数据表名称(技术报告有效)。
start_timestamp	否	Long	开始时间戳。
end_timestamp	否	Long	结束时间戳。

## 请求参数

表 9-113 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-114 响应 Body 参数

参数	参数类型	描述
score	Double	总分。
ranges	Array of <a href="#">ScoreRangeVO</a> objects	区间列表及个数。

表 9-115 ScoreRangeVO

参数	参数类型	描述
count	Integer	个数。
name	String	区间名。

状态码： 400

表 9-116 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500



表 9-117 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

```
GET /v2/0833a5737480d53b2f25c010dc1a7b88/quality/report/overview?
data_connection_id=2d9dcb2076b34bbab1c675f070d6af9d&quality_score_dimension=technology
```

## 响应示例

状态码： 200

Success

```
{
  "score": 56.75,
  "ranges": [ {
    "count": 9,
    "name": "80-100"
  }, {
    "count": 2,
    "name": "60-80"
  }, {
    "count": 2,
    "name": "40-60"
  }, {
    "count": 6,
    "name": "0-20"
  } ]
}
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.7.3 获取质量报告趋势

### 功能介绍

获取质量报告趋势。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/quality/report/trend

表 9-118 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-119 Query 参数

参数	是否必选	参数类型	描述
quality_score_dimension	是	String	质量评分维度。
l1	否	String	主题域分组(业务报告有效)。
l2	否	String	主题域(业务报告有效)。
l3	否	String	业务对象(业务报告有效)。
data_connection_id	否	String	数据连接id(技术报告有效)。
database_name	否	String	数据库名称(技术报告有效)。
table_name	否	String	数据表名称(技术报告有效)。
start_timestamp	否	Long	开始时间戳。
end_timestamp	否	Long	结束时间戳。

## 请求参数

表 9-120 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-121 响应 Body 参数

参数	参数类型	描述
trends	Array of <b>ScoreTrendVO</b> objects	分数时间变化趋势数组ScoreTrendVO。

表 9-122 ScoreTrendVO

参数	参数类型	描述
score	Double	分数。
timestamp	String	时间戳。

状态码： 400

表 9-123 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-124 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

```
GET /v2/0833a5737480d53b2f25c010dc1a7b88/quality/report/trend?
data_connection_id=2d9dcb2076b34bbab1c675f070d6af9d&quality_score_dimension=technology
```

## 响应示例

状态码: 200

Success

```
{
  "trends": [ {
    "score": 57.63,
    "timestamp": "1669132800000"
  }, {
    "score": 56.92,
    "timestamp": "1669219200000"
  }, {
    "score": 60.27,
    "timestamp": "1669305600000"
  }, {
    "score": 59.54,
    "timestamp": "1669392000000"
  }, {
    "score": 65.07,
    "timestamp": "1669478400000"
  }, {
    "score": 57.19,
    "timestamp": "1669564800000"
  }, {
    "score": 56.75,
    "timestamp": "1669651200000"
  }
]
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.7.4 获取质量报告规则评分

### 功能介绍

获取质量报告规则评分。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/report/rules

表 9-125 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-126 Query 参数

参数	是否必选	参数类型	描述
data_connection_id	否	String	数据连接id, 如果是获取技术报告规则评分, 请务必填写 data_connection_id, database_name, table_name 参数值。
database_name	否	String	数据库名称, 如果是获取技术报告规则评分, 请务必填写 data_connection_id, database_name, table_name 参数值。
table_name	否	String	表名称, 如果是获取技术报告规则评分, 请务必填写 data_connection_id, database_name, table_name 参数值。
table_object_name	否	String	数据表对象名 (数据连接.库.模式表/表), 如果是获取业务报告规则评分, data_connection_id, database_name, table_name 参数值不需要填写, 只需要填写 table_object_name 参数值。
start_timestamp	否	Long	开始时间戳。
end_timestamp	否	Long	结束时间戳。
limit	否	Long	分页条数,取值范围[0,100]。
offset	否	Long	分页偏移量,最小值0。

## 请求参数

表 9-127 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-128 响应 Body 参数

参数	参数类型	描述
count	Integer	数据条数。
rules	Array of <a href="#">BusinessScoreRuleVO</a> objects	返回数据列表BusinessScoreRuleVO

表 9-129 BusinessScoreRuleVO

参数	参数类型	描述
rule_name	String	规则名。
type	Integer	规则类型，0表示自定义，1表示系统。
score	Double	规则得分。
rule_id	String	规则ID，用于跳转规则详情页。
instance_id	String	规则实例ID，用于跳转结果页。
scoring_weight	Integer	规则权重。
rule_type_name	String	规则类型名称。
dimension_name	String	规则维度。
sub_rule_instance_id	String	子规则实例ID。

状态码： 400

表 9-130 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-131 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

```
GET /v2/0833a5737480d53b2f25c010dc1a7b88/quality/report/rules?
data_connection_id=2d9dcb2076b34bbab1c675f070d6af9d&database_name=postgres&table_name=public.te
st
```

## 响应示例

状态码： 200

Success

```
{
  "count": 1,
  "rules": [ {
    "rule_name": "test",
    "type": 0,
    "score": 0.0,
    "rule_id": "1044554894322835457",
    "instance_id": "1047298515736334336",
    "scoring_weight": 5,
    "rule_type_name": "数据表空值扫描",
    "dimension_name": "完整性",
    "sub_rule_instance_id": "1047298617787944960"
  } ]
}
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.7.5 获取质量报告子规则字段评分

### 功能介绍

获取质量报告子规则字段评分。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/report/sub-rule/results

表 9-132 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-133 Query 参数

参数	是否必选	参数类型	描述
data_connection_id	否	String	数据连接id，如果是获取技术报告规则评分，请务必填写 data_connection_id, database_name, table_name 参数值。
database_name	否	String	数据库名称，如果是获取技术报告规则评分，请务必填写 data_connection_id, database_name, table_name 参数值。
table_name	否	String	表名称，如果是获取技术报告规则评分，请务必填写 data_connection_id, database_name, table_name 参数值。



参数	是否必选	参数类型	描述
table_object_name	否	String	数据表对象名（数据连接.库.模式表/表）。如果是获取业务报告规则评分，data_connection_id, database_name, table_name 参数值不需要填写，只需要填写 table_object_name 参数值。
sub_rule_instance_id	是	String	子规则实例ID。
start_timestamp	否	Long	开始时间戳。
end_timestamp	否	Long	结束时间戳。
limit	否	Long	分页条数,取值范围[0,100]。
offset	否	Long	分页偏移量,最小值0。

## 请求参数

表 9-134 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio 工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token, Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-135 响应 Body 参数

参数	参数类型	描述
header	Array of strings	表头内容。
rows	Array<Array<String>>	表行内容。
count	Integer	数据总行数，用于分页。

状态码： 400

表 9-136 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-137 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

```
GET /v2/0833a5737480d53b2f25c010dc1a7b88/quality/report/sub-rule/results?
data_connection_id=2d9dcb2076b34bbab1c675f070d6af9d&database_name=postgres&table_name=public.te
st&sub_rule_instance_id=1047298617787944960
```

## 响应示例

状态码： 200

Success

```
{
  "count": 1,
  "header": [ "字段名称", "规则描述", "分数", "字段权重", "空值行数", "总行数", "空值率", "告警状态" ],
  "rows": [ [ "postgres.public.test.name", "数据表空值扫描", "0.0", "5", "0", "0", "1.0", "false" ] ]
}
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.7.6 获取质量报告技术报告数据

### 功能介绍

获取质量报告技术报告数据。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/report/technology/scores

表 9-138 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-139 Query 参数

参数	是否必选	参数类型	描述
data_connection_id	否	String	数据连接id，通过管理中心获取连接ID。
database_name	否	String	数据库名称。
table_name	否	String	表名称。
data_connection_score_order	否	String	数据连接评分排序，0表示升序，1表示降序，与其他排序条件互斥。
database_score_order	否	String	数据库评分排序，0表示升序，1表示降序，与其他排序条件互斥。
table_score_order	否	String	数据表评分排序，0表示升序，1表示降序，与其他排序条件互斥。
start_timestamp	否	Long	开始时间戳。
end_timestamp	否	Long	结束时间戳。
limit	否	Long	分页条数,取值范围[0,100]。

参数	是否必选	参数类型	描述
offset	否	Long	分页偏移量,最小值0。

## 请求参数

表 9-140 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token, Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-141 响应 Body 参数

参数	参数类型	描述
count	Integer	数据条数。
scores	Array of <a href="#">TechScoreVo</a> objects	返回数据列表TechnologyScoreVO。

表 9-142 TechScoreVo

参数	参数类型	描述
data_connection_id	String	数据连接ID。
data_connection_name	String	数据连接名称。
database_name	String	数据库名称。
table_name	String	数据表名称。
data_connection_score	Double	数据连接评分。
database_score	Double	数据库评分。

参数	参数类型	描述
table_score	Double	表评分。

状态码： 400

表 9-143 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-144 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

```
GET /v2/0833a5737480d53b2f25c010dc1a7b88/quality/report/technology/scores
```

## 响应示例

状态码： 200

Success

```
{
  "count": 1,
  "scores": [ {
    "data_connection_id": "2d9dcb2076b34bbab1c675f070d6af9d",
    "data_connection_name": "dws",
    "data_connection_score": 65.0,
    "database_name": "postgres",
    "database_score": 65.0,
    "table_name": "public.test",
    "table_score": 0.0
  } ]
}
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.7.7 获取质量报告业务报告数据

### 功能介绍

获取质量报告业务报告数据。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/report/business/scores

表 9-145 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-146 Query 参数

参数	是否必选	参数类型	描述
table_name	否	String	表名称。
l1	否	String	主题域分组。
l2	否	String	主题域。
l3	否	String	业务对象。
l1_score_order	否	String	主题域分组排序，0表示升序，1表示降序，与其他排序条件互斥。
l2_score_order	否	String	主题域排序，0表示升序，1表示降序，与其他排序条件互斥。

参数	是否必选	参数类型	描述
l3_score_order	否	String	业务对象排序, 0表示升序, 1表示降序, 与其他排序条件互斥。
table_score_order	否	String	数据表评分排序, 0表示升序, 1表示降序, 与其他排序条件互斥。
start_timestamp	否	Long	开始时间戳。
end_timestamp	否	Long	结束时间戳。
limit	否	Long	分页条数, 取值范围[0,100]。
offset	否	Long	分页偏移量, 最小值0。

## 请求参数

表 9-147 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token, Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码: 200

表 9-148 响应 Body 参数

参数	参数类型	描述
count	Integer	数据条数。
scores	Array of <a href="#">BusinessScoreVO</a> objects	返回数据列表BusinessScoreVO。

表 9-149 BusinessScoreVO

参数	参数类型	描述
id	String	评分编号。
l1	String	主题域分组。
l2	String	主题域。
l3	String	业务对象。
table_name	String	数据表对象名，格式为：数据连接.库.模式表/表。
l1_score	Double	主题域分组评分。
l2_score	Double	主题域评分。
l3_score	Double	业务对象评分。
table_score	Double	表评分。

状态码： 400

表 9-150 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

状态码： 500

表 9-151 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

```
GET /v2/0833a5737480d53b2f25c010dc1a7b88/quality/report/business/scores
```

## 响应示例

状态码： 200



Success

```
{
  "count": 1,
  "scores": [ {
    "id": "1047319323363946496",
    "l1": "test",
    "l2": null,
    "l3": null,
    "table_name": "2d9dcb2076b34bbab1c675f070d6af9d.postgres.public.test",
    "l1_score": 83.33,
    "l2_score": null,
    "l3_score": null,
    "table_score": 83.33
  } ]
}
```

## 状态码

状态码	描述
200	Success
400	BadRequest
500	INTERNAL SERVER ERROR

## 9.8 导入导出接口

### 9.8.1 导出资源

#### 功能介绍

该接口用于数据质量导出操作，可通过该接口导出质量作业或者对账作业等属于数据质量服务的资源数据。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v2/{project\_id}/quality/resource/export

表 9-152 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 9-153 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-154 请求 Body 参数

参数	是否必选	参数类型	描述
type	是	String	批量导出的资源类型：rule-template表示质量规则模板，quality-task表示质量作业，consistency-task表示对账作业，report表示质量报告。
is_select_all	否	Boolean	是否导出当前空间的所有作业，true表示导出所有，false表示否，默认 false。
resources	否	Array of longs	资源ID数组。
export_mode	否	String	导出模式，null表示导出到本地文件系统，obs表示导出至obs，仅限于导出质量报告使用。
obs_service_type	否	String	业务类型，technology表示技术报告，business表示业务报告，仅限于导出质量报告使用。

## 响应参数

状态码： 200

表 9-155 响应 Body 参数

参数	参数类型	描述
resource_id	String	返回的导出任务ID。

状态码： 500

表 9-156 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

### 请求示例

无

### 响应示例

无

### 状态码

状态码	描述
200	Success
500	INTERNAL SERVER ERROR

## 9.8.2 获取导入导出的任务状态

### 功能介绍

该接口用于获取导出导入任务的进度状态。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/resource/status/{resource\_id}

表 9-157 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
resource_id	是	String	资源导出导入任务ID，通过 <a href="#">导出资源</a> 或者 <a href="#">导入资源</a> 返回内容获得。

## 请求参数

表 9-158 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-159 响应 Body 参数

参数	参数类型	描述
id	String	记录ID。
resource_id	String	资源导出导入任务ID。
resource_type	String	资源类型。
action	String	操作类型：EXPORT表示导出，REPORT表示导入。
file_name	String	资源文件名。
start_time	String	开始时间。
end_time	String	结束时间。
status	String	任务状态，WAITING表示等待执行，RUNNING表示执行中，TIMEOUT表示执行超时，FAILED表示执行失败，SUCCESS表示执行成功。
error_msg	String	错误信息。
project_id	String	项目ID。
domain_id	String	用户ID。
user_name	String	用户名。
export_resources	String	导出资源信息，默认为空。
import_body	String	导入body体，默认为空。

参数	参数类型	描述
file_status	String	文件进度状态，PREPARING表示准备中，GENERATED表示文件已就绪，DELETED表示文件已删除，FAILED表示上传或生成文件失败。

状态码： 500

表 9-160 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

无

## 状态码

状态码	描述
200	Success
500	INTERNAL SERVER ERROR

## 9.8.3 下载资源任务文件

### 功能介绍

该接口用于下载资源文件。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/quality/resource/download

表 9-161 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 9-162 Query 参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源导出导入任务ID，通过 <a href="#">导出资源</a> 或者 <a href="#">导入资源</a> 返回内容获得。
obs_type	否	String	导出类型，技术报告：technology，业务报告：business，当导出质量报告时，该字段是必填项。

## 请求参数

表 9-163 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

## 响应参数

状态码： 200

表 9-164 响应 Body 参数

参数	参数类型	描述
-	File	下载的资源文件。

状态码： 500

表 9-165 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

### 请求示例

无

### 响应示例

无

### 状态码

状态码	描述
200	下载的资源文件。
500	INTERNAL SERVER ERROR

## 9.8.4 上传导入的文件接口

### 功能介绍

该接口用于数据质量导入文件使用。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/quality/resource/upload

表 9-166 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 9-167 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-168 FormData 参数

参数	是否必选	参数类型	描述
type	是	String	需要上传的资源type (rule-template、quality-task、consistency-task)。
file	是	File	上传的资源文件。

## 响应参数

状态码： 200

表 9-169 响应 Body 参数

参数	参数类型	描述
resource_id	String	资源ID。
topics	Array of strings	smn主题。
matched_topics	Array of <a href="#">MatchedTopicMs g</a> objects	匹配到的主题信息。
directories	Array of strings	数据质量目录信息。
queues	Array of strings	队列名称。
connections	Array of <a href="#">NameType</a> objects	数据连接信息。



表 9-170 MatchedTopicMsg

参数	参数类型	描述
id	String	主题ID。
name	String	主题名称。

表 9-171 NameType

参数	参数类型	描述
name	String	名称。
type	String	类型。

状态码： 500

表 9-172 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

无

## 状态码

状态码	描述
200	Success
500	INTERNAL SERVER ERROR

## 9.8.5 导入资源

### 功能介绍

该接口用于数据质量导入操作使用。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/quality/resource/import

表 9-173 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 9-174 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	DataArts Studio工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，Token获取请参见 <a href="#">认证鉴权</a> 。

表 9-175 请求 Body 参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源ID。
type	是	String	需要导入的资源类型：rule-template表示质量规则模板，quality-task表示质量作业，consistency-task表示对账作业。

参数	是否必选	参数类型	描述
strategy	否	String	导入出现同名资源时处理策略：SKIP表示忽略、OVERRIDE表示覆盖、TERMINATION表示终止。
directories	否	Array of ValueMappingVO objects	目录调整变化。
topics	否	Array of ValueMappingVO objects	告警通知主题调整。
connections	否	Array of ValueMappingVO objects	连接调整，只针对DLI。
queues	否	Array of ValueMappingVO objects	队列调整，只针对DLI。

表 9-176 ValueMappingVO

参数	是否必选	参数类型	描述
old_value	否	String	映射前数据。
new_value	否	String	映射后数据。
mapping_value	否	String	映射后数据, 可以为空, 为空则使用new_value。

## 响应参数

状态码： 500

表 9-177 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码，如DQC.0000表示请求处理成功。
error_msg	String	错误信息。

## 请求示例

无

## 响应示例

无

## 状态码

状态码	描述
204	Success
500	INTERNAL SERVER ERROR

# 10 数据目录 API

## 10.1 业务资产接口

### 10.1.1 查询业务资产

#### 功能介绍

查询业务资产，包含数据规范同步过来的业务对象和逻辑实体。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/asset/business-assets/search

表 10-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-3 请求 Body 参数

参数	是否必选	参数类型	描述
search_all_attributes	是	Boolean	关键字查询是否匹配所有属性，true为查询所有属性，false为仅查询名称描述。
tags	否	Object	标签列表。
limit	是	Integer	查询返回数目。
offset	是	Integer	查询偏移量。
guid	否	String	查询节点的guid。
query	是	String	查询关键字，search_all_attributes为true查询所有属性，false仅查询名称描述。
type	是	String	查询类型。

## 响应参数

状态码： 200

表 10-4 响应 Body 参数

参数	参数类型	描述
count	Integer	业务资产总数。
assets	Array of <a href="#">OpenEntity</a> objects	业务资产列表。

表 10-5 OpenEntity

参数	参数类型	描述
attributes	Object	属性Map(String, Object), key: 属性名称, value: 属性值。
connection	<b>Connection</b> object	数据连接。
create_time	Number	创建时间。
created_by	String	创建人。
display_text	String	资产的名称。
guid	String	资产guid。
relationship_attributes	Object	相关的属性 Map(String, Object), key: 关联关系类型, value: 关联关系信息。
type_name	String	资产类型。
updated_by	String	更新人。
update_time	Number	更新时间。
tags	Array of <b>TagHeader</b> objects	标签。
classification_names	Array of strings	分类名称列表。

表 10-6 Connection

参数	参数类型	描述
guid	String	关联guid。
display_text	String	显示内容。
type_name	String	类型名称。
connection_type	String	连接类型。
qualified_name	String	限定名称。

表 10-7 TagHeader

参数	参数类型	描述
name	String	资产名称。
dexcription	Object	标签描述。
display_text	String	标签的名称。
relation_guid	String	关联的guid。
tag_guid	String	标签关联的guid。

状态码： 400

表 10-8 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-9 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-10 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404



表 10-11 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "search_all_attributes" : false,
  "limit" : 10,
  "offset" : 0,
  "query" : "dli_table",
  "type" : "LOGICENTITY"
}
```

## 响应示例

状态码： 200

Business assets。

```
{
  "assets" : [ {
    "attributes" : {
      "owner" : "",
      "code" : "53f31e71-512a-41d8-81d8-7c49ce68d7b3",
      "description" : "无",
      "updateTime" : 1661908856018,
      "dataClassify" : "BASE_DATA",
      "securityLevel" : null,
      "nameEng" : "dli_table",
      "tables" : [ {
        "uniqueAttributes" : {
          "qualifiedName" : "cbu_training.dli_table@dli.0833a5737480d53b2f25c010dc1a7b88-workspace-61aa10df45e54431a1901cb3527adab8"
        },
        "type_name" : "dli_table_managed",
        "name" : "dli_table",
        "guid" : "d99a8270-63b5-4252-ad37-8cebe7530a73"
      } ],
      "selfDefinedFields" : null,
      "qualified_name" : "subject_area_group.subject_area.object.dli_table@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-61aa10df45e54431a1901cb3527adab8",
      "createTime" : 1661908856018,
      "name" : "dli_table",
      "alias" : null,
      "fields" : [ {
        "type_name" : "BusinessLogicEntityColumn",
        "guid" : "d05187d0-adb0-4953-8239-eedb88c21b30"
      }, {
        "type_name" : "BusinessLogicEntityColumn",
        "guid" : "d75821b5-ef73-4650-ac4e-340b255db412"
      }, {
        "type_name" : "BusinessLogicEntityColumn",
        "guid" : "cd7e8939-50f7-4096-b4e4-9f7b73167a97"
      } ],
      "parameters" : null,
      "publishStatus" : null,
      "workspaceId" : "61aa10df45e54431a1901cb3527adab8"
    },
    "classification_names" : null,
  } ]
}
```

```
"connection" : null,
"create_time" : 1661908450270,
"created_by" : "username",
"display_text" : "dli_table",
"guid" : "a970c4fb-ac97-4339-95e6-944912c58a2b",
"relationship_attributes" : {
  "tables" : [ {
    "display_text" : null,
    "guid" : "d99a8270-63b5-4252-ad37-8cebe7530a73",
    "relationship_attributes" : null,
    "relationship_guid" : null,
    "type_name" : null
  } ],
  "catelog" : {
    "display_text" : null,
    "guid" : "7fbed9da-331d-4695-b8fa-5587605f37e4",
    "relationship_attributes" : null,
    "relationship_guid" : null,
    "type_name" : null
  },
  "fields" : [ {
    "display_text" : null,
    "guid" : "d05187d0-adb0-4953-8239-eedb88c21b30",
    "relationship_attributes" : null,
    "relationship_guid" : null,
    "type_name" : null
  }, {
    "display_text" : null,
    "guid" : "d75821b5-ef73-4650-ac4e-340b255db412",
    "relationship_attributes" : null,
    "relationship_guid" : null,
    "type_name" : null
  }, {
    "display_text" : null,
    "guid" : "cd7e8939-50f7-4096-b4e4-9f7b73167a97",
    "relationship_attributes" : null,
    "relationship_guid" : null,
    "type_name" : null
  } ],
  "tags" : [ ]
},
"tags" : null,
"type_name" : "BusinessLogicEntity",
"update_time" : 1661908450270,
"updated_by" : "uaername"
} ],
"count" : 1
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowBusinessAssetsSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowBusinessAssetsRequest request = new ShowBusinessAssetsRequest();
    BusinessAssetRequest body = new BusinessAssetRequest();
    request.withBody(body);
    try {
        ShowBusinessAssetsResponse response = client.showBusinessAssets(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowBusinessAssetsRequest()
```

```

request.body = BusinessAssetRequest(
)
response = client.show_business_assets(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBusinessAssetsRequest{}
    request.Body = &model.BusinessAssetRequest{
    }
    response, err := client.ShowBusinessAssets(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Business assets。

状态码	描述
400	Bad Request。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.1.2 查询业务资产目录树

### 功能介绍

逐级查询业务资产目录树，包含数据规范同步过来的业务对象和逻辑实体。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/business-assets/tree/subnode

表 10-12 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 10-13 Query 参数

参数	是否必选	参数类型	描述
guid	否	String	资产guid，未填充时查询LV1级别业务资产，获取方法请参见 <a href="#">数据资产guid</a> 。

## 请求参数

表 10-14 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-15 响应 Body 参数

参数	参数类型	描述
business_catalog_guid	String	业务资产guid。
business_catalog_name	String	业务资产名称。
business_catalog_name_eng	String	业务资产英文名称。
level	String	业务资产级别。
qualified_name	String	业务资产级唯一限定名称。
ordinal	Integer	序数。
child_nodes	Array of <a href="#">BusinessCatalogTreeNode</a> objects	子级业务资产列表。
logic_entity_names	Array of <a href="#">LogicEntityNames</a> objects	逻辑实体列表。

表 10-16 LogicEntityNodes

参数	参数类型	描述
logic_entity_guid	String	业务资产guid。
logic_entity_name	String	业务资产名称。

状态码： 400

表 10-17 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-18 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-19 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-20 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

Business assets tree。

```
{
  "business_catalog_name": "ZTY",
  "business_catalog_name_eng": "ZTY",
  "qualified_name": "ZTY@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-1b59d3c777ad4d619b89eeac4f3cce87",
  "logic_entity_nodes": [ {
    "logic_entity_name": "luojiyi",
    "logic_entity_guid": "b8ef87fb-e190-4c10-815e-9b43dae366b2"
  } ],
  "level": "L1",
  "child_nodes": [ {
    "business_catalog_name": "zty_l2",
    "business_catalog_name_eng": "zty_l2",
    "qualified_name": "ZTY.zty_l2@Business.0833a5737480d53b2f25c010dc1a7b88-workspace-1b59d3c777ad4d619b89eeac4f3cce87",
    "logic_entity_nodes": [ ],
    "level": "L2",
    "child_nodes": [ ],
    "business_catalog_guid": "9b5cab1f-be30-47f9-9138-4427e808daa3",
    "ordinal": 1
  } ],
  "business_catalog_guid": "5ae65812-d0a6-43a7-876a-c43561e92f3b",
  "ordinal": 470
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowBusinessAssetsTreeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```



```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowBusinessAssetsTreeRequest request = new ShowBusinessAssetsTreeRequest();
try {
    ShowBusinessAssetsTreeResponse response = client.showBusinessAssetsTree(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowBusinessAssetsTreeRequest()
        response = client.show_business_assets_tree(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBusinessAssetsTreeRequest{}
    response, err := client.ShowBusinessAssetsTree(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Business assets tree。
400	Bad Request。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.2 指标资产接口

### 10.2.1 查询指标资产目录树

#### 功能介绍

查询指标资产目录树。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{project\_id}/asset/metric-tree

表 10-21 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 10-22 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

#### 响应参数

状态码： 200

表 10-23 响应 Body 参数

参数	参数类型	描述
architecture	Array of <b>ArchitectureStatistic</b> objects	结构体系。

表 10-24 ArchitectureStatistic

参数	参数类型	描述
children	Array of <b>ArchitectureStatistic</b> objects	子指标。
count	Integer	子指标个数。
guid	String	资产guid。
name	String	名称。

状态码： 401

表 10-25 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-26 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-27 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

metric assets。

```
{
  "architecture": [ {
    "children": [ {
      "children": null,
      "count": 1,
      "guid": "f31ad3b7-5c54-45e3-8afe-369c43d45438",
      "name": "process_name_second"
    } ],
    "count": 0,
    "guid": "64db45a2-38d1-453d-bb13-404dde03fedc",
    "name": "process_name_first"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowMetricTreeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowMetricTreeRequest request = new ShowMetricTreeRequest();
try {
    ShowMetricTreeResponse response = client.showMetricTree(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMetricTreeRequest()
        response = client.show_metric_tree(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
```

```

"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMetricTreeRequest{}
    response, err := client.ShowMetricTree(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	metric assets。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.2.2 查询指标资产

### 功能介绍

指标资产查询接口。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/{project\_id}/asset/metric-assets/search

表 10-28 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-29 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-30 请求 Body 参数

参数	是否必选	参数类型	描述
architecture_guid	否	String	指标资产ID。
query	是	String	查询条件，search_name_description为true查询按照指标名称或者描述，为false时，查询按照其他指标属性。
limit	否	Integer	单次请求条数。
offset	否	Integer	偏移量。
search_name_description	否	Boolean	是否按名称和描述搜索。
include_sub_architecture	否	Boolean	是否查询子指标。



## 响应参数

状态码： 200

表 10-31 响应 Body 参数

参数	参数类型	描述
count	Integer	指标资产总数。
entities	Array of <b>OpenEntityHeader</b> objects	指标资产列表。
scroll_id	String	scroll_id

表 10-32 OpenEntityHeader

参数	参数类型	描述
attributes	Object	属性。
connection	<b>Connection</b> object	数据连接。
display_text	String	展示文档。
guid	String	资产guid。
type_name	String	类型名称。
tags	Array of <b>TagHeader</b> objects	标签列表。
classification_names	Array of strings	分类名称列表。

表 10-33 Connection

参数	参数类型	描述
guid	String	关联guid。
display_text	String	显示内容。
type_name	String	类型名称。
connection_type	String	连接类型。
qualified_name	String	限定名称。

表 10-34 TagHeader

参数	参数类型	描述
name	String	资产名称。
dexcription	Object	标签描述。
display_text	String	标签的名称。
relation_guid	String	关联的guid。
tag_guid	String	标签关联的guid。

状态码： 401

表 10-35 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-36 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-37 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{  
  "query" : "indicator",  
  "search_name_description" : true  
}
```

## 响应示例

状态码: 200

metric assets。

```
{
  "count" : 1,
  "entities" : [ {
    "attributes" : {
      "owner" : "username",
      "path" : "/process_name_first/process_name_second",
      "create_time" : 1661910746000,
      "qualified_name" :
"1014471843710640128.1014471946009714688.1014472726615900160@Business.0833a5737480d53b2f25c
010dc1a7b88-workspace-61aa10df45e54431a1901cb3527adab8",
      "name" : "indicator",
      "description" : null,
      "definition" : "Indicator Example",
      "security_level" : null
    },
    "classification_names" : [ ],
    "connection" : null,
    "display_text" : "indicator",
    "guid" : "2fd90dc8-f130-47c2-b6a6-141761c4f9f4",
    "tags" : [ ],
    "type_name" : "BusinessMetric"
  } ],
  "scroll_id" : null
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowMetricAssetsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowMetricAssetsRequest request = new ShowMetricAssetsRequest();
    MetricOpenSearchParams body = new MetricOpenSearchParams();
    request.withBody(body);
    try {
        ShowMetricAssetsResponse response = client.showMetricAssets(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMetricAssetsRequest()
        request.body = MetricOpenSearchParams(
        )
        response = client.show_metric_assets(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```

dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMetricAssetsRequest{}
    request.Body = &model.MetricOpenSearchParams{
    }
    response, err := client.ShowMetricAssets(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	metric assets。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.3 统计资产接口

## 10.3.1 获取技术资产统计信息

### 功能介绍

获取技术资产统计信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/asset/statistic/assets/technical-assets

表 10-38 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 10-39 Query 参数

参数	是否必选	参数类型	描述
tag	否	String	标签名，指定标签名称可以获取此标签对应技术资产的统计信息。
offset	否	Integer	分页参数，查询偏移量，默认查询所有。
limit	否	Integer	分页参数，每页数量，默认查询所有。

### 请求参数

表 10-40 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-41 响应 Body 参数

参数	参数类型	描述
count	Integer	数据连接总数。
datasource_statistics	Array of <b>DataSource</b> objects	数据连接统计信息。

表 10-42 DataSource

参数	参数类型	描述
datasource_name	String	数据连接名称。
datasource_type	String	数据连接类型。
datasource_guid	String	数据连接guid。
datasource_qualified_name	String	数据连接唯一标识名称。
obs_folder_count	Integer	obs目录数。
obs_file_count	Integer	obs文件数。
css_index_count	Integer	css索引数。
css_index_field_count	Integer	css 索引字段数目。
namespace_count	Integer	命名空间数。
ges_vertex_count	Integer	ges点的总数。
ges_edge_count	Integer	ges边的总数。
database_count	Integer	数据库总数。
stream_count	Integer	通道总数。
table_count	Integer	表总数。

参数	参数类型	描述
data_size	Double	数据大小。
databases	Array of <b>Database</b> objects	数据库统计信息。
folders	Array of <b>ObsFolder</b> objects	顶层目录统计信息。
css_indices	Array of <b>CssIndex</b> objects	css索引统计信息。
namespaces	Array of <b>Namespace</b> objects	命名空间统计信息。
dis_streams	Array of <b>DisStream</b> objects	通道统计信息。

表 10-43 Database

参数	参数类型	描述
database_name	String	数据库名称。
database_guid	String	数据库guid。
database_qualified_name	String	数据库的唯一标识名称。
table_count	Integer	数据库中表数目。
data_size	Double	数据量大小。

表 10-44 ObsFolder

参数	参数类型	描述
folder_name	String	目录名称。
folder_guid	String	目录的guid。
folder_qualified_name	String	目录的唯一标识名称。
object_count	Integer	对象总数。



参数	参数类型	描述
data_size	Double	数据量。

表 10-45 CssIndex

参数	参数类型	描述
index_name	String	索引名称。
index_guid	String	索引的guid。
index_qualified_name	String	索引的唯一标识名称。
index_doc_count	Integer	索引中文档总数。
index_data_size	Double	索引数据量大小。

表 10-46 Namespace

参数	参数类型	描述
namespace_name	String	命名空间的名称。
namespace_guid	String	命名空间的guid。
namespace_qualified_name	String	命名空间的唯一标识名称。
table_count	Integer	命名空间下的表总数。

表 10-47 DisStream

参数	参数类型	描述
stream_name	String	通道名称。
stream_guid	String	通道GUID。
stream_qualified_name	String	通道的唯一标识名称。
partition_count	Integer	分区数。
app_count	Integer	dis的app数目。

参数	参数类型	描述
task_count	Integer	转储任务数。

状态码： 400

表 10-48 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-49 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-50 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-51 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码: 200

Technical assets statistic。

```
{
  "count": 1,
  "datasource_statistics": [ {
    "datasource_name": "dli",
    "datasource_type": "dli",
    "datasource_guid": "0fff4057-c206-4dc3-a4ac-73ffc332bc9a",
    "datasource_qualified_name": "dli@0833a5737480d53b2f25c010dc1a7b88-
workspace-61aa10df45e54431a1901cb3527adab8",
    "obs_folder_count": 0,
    "obs_file_count": 0,
    "css_index_count": 0,
    "css_index_field_count": 0,
    "namespace_count": 0,
    "ges_vertex_count": 0,
    "ges_edge_count": 0,
    "database_count": 1,
    "stream_count": 0,
    "table_count": 1,
    "data_size": 5478,
    "databases": [ {
      "data_size": 5478,
      "database_guid": "e2b12c35-48ee-441f-a357-e338a85f5d00",
      "database_name": "cbu_training",
      "database_qualified_name": "cbu_training@dli.0833a5737480d53b2f25c010dc1a7b88-
workspace-61aa10df45e54431a1901cb3527adab8",
      "table_count": 1
    } ],
    "folders": null,
    "css_indices": null,
    "namespaces": null,
    "dis_streams": null
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowTechnicalAssetsStatisticSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowTechnicalAssetsStatisticRequest request = new ShowTechnicalAssetsStatisticRequest();
try {
    ShowTechnicalAssetsStatisticResponse response = client.showTechnicalAssetsStatistic(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowTechnicalAssetsStatisticRequest()
        response = client.show_technical_assets_statistic(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowTechnicalAssetsStatisticRequest{}
    response, err := client.ShowTechnicalAssetsStatistic(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Technical assets statistic。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.3.2 获取业务资产统计信息

### 功能介绍

获取业务资产统计信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/asset/statistic/assets/business-assets

表 10-52 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 10-53 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页参数，查询偏移量，默认查询所有。
limit	否	Integer	分页参数，每页数量，默认查询所有。

### 请求参数

表 10-54 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-55 响应 Body 参数

参数	参数类型	描述
count	Integer	主题域分组的总数。
subject_area_group_statistics	Array of <b>L1Statistic</b> objects	主题域分组的统计信息。

表 10-56 L1Statistic

参数	参数类型	描述
subject_area_group_name	String	主题域分组名称。
subject_area_group_name_en	String	主题域分组英文名称。
subject_area_group_guid	String	主题域分组的guid。
ordinal	Integer	主题域分组序号。
subject_area_count	Integer	主题总数。
business_object_count	Integer	业务对象总数。
logic_entity_count	Integer	逻辑实体总数。
subject_area_statistics	Array of <b>L2Statistic</b> objects	主题统计信息。

表 10-57 L2Statistic

参数	参数类型	描述
subject_area_name	String	主题名称。
subject_area_guid	String	主题的guid。

参数	参数类型	描述
ordinal	Integer	主题序号。
business_object_count	Integer	业务对象总数。
logic_entity_count	Integer	逻辑实体总数。

状态码： 400

表 10-58 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-59 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-60 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404



表 10-61 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

Business assets statistic。

```
{
  "count" : 1,
  "subject_area_group_statistics" : [ {
    "subject_area_group_name" : "subject_area_group",
    "subject_area_group_name_en" : "subject_area_group",
    "subject_area_group_guid" : "ea60986c-7de3-4743-9ae3-2b3d8b030366",
    "ordinal" : 1,
    "subject_area_count" : 1,
    "business_object_count" : 1,
    "logic_entity_count" : 1,
    "subject_area_statistics" : [ {
      "subject_area_name" : "subject_area",
      "subject_area_guid" : "e80d6b75-83a9-40aa-b621-59188690d28a",
      "business_object_count" : 1,
      "logic_entity_count" : 1,
      "ordinal" : 1
    }
  ]
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowBusinessAssetsStatisticSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowBusinessAssetsStatisticRequest request = new ShowBusinessAssetsStatisticRequest();
try {
    ShowBusinessAssetsStatisticResponse response = client.showBusinessAssetsStatistic(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowBusinessAssetsStatisticRequest()
        response = client.show_business_assets_statistic(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBusinessAssetsStatisticRequest{}
    response, err := client.ShowBusinessAssetsStatistic(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Business assets statistic。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found

## 10.4 资产管理接口

### 10.4.1 查询技术资产

#### 功能介绍

查询技术资产。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/asset/entities/search

表 10-62 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 10-63 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-64 请求 Body 参数

参数	是否必选	参数类型	描述
query	是	String	查询关键字，search_all_attributes为true匹配所有属性，false只匹配名称、描述，默认false。

参数	是否必选	参数类型	描述
type_names	是	Array of strings	分类名称列表。
connection_names	否	Array of strings	连接名称列表。
search_all_attributes	否	Boolean	查询关键字是否匹配资产的名称描述信息，true：匹配所有属性，false：只匹配名称、描述，默认false。
tags	否	Array of strings	标签名称列表。
limit	是	Integer	分页显示每页返回结果数。默认值：10。
offset	否	Integer	偏移量，默认值：0。
attributes	否	Object	key当前支持Table，value可为以下中的一个或多个： rowCounts、tableSize、database、schema、namespace、ddlUpdateTime、dataUpdateTime、ddlCreateTime Map(String,Set(String))。
filter_criteria	否	<b>FilterCriteria</b> object	过滤条件。
time_range	否	<b>TimeRange</b> object	时间过滤条件，可以根据此条件设置过滤资产。
scroll_id	否	String	scroll_id。
security_levels	否	Array of strings	密级列表。
is_import	否	Boolean	是否导入。
classifications	否	Array of strings	分类列表。
description	否	String	描述。

表 10-65 FilterCriteria

参数	是否必选	参数类型	描述
name	是	String	当前可选值：database。

参数	是否必选	参数类型	描述
value	是	String	database的名称。
operator	是	String	操作标识。

表 10-66 TimeRange

参数	是否必选	参数类型	描述
time_type	是	String	时间类型。
start	是	String	开始时间。
end	是	String	结束时间。

## 响应参数

状态码： 200

表 10-67 响应 Body 参数

参数	参数类型	描述
count	Integer	技术资产总数。
entities	Array of <b>OpenEntityHeader</b> objects	技术资产列表。
scroll_id	String	scroll_id

表 10-68 OpenEntityHeader

参数	参数类型	描述
attributes	Object	属性。
connection	<b>Connection</b> object	数据连接。
display_text	String	展示文档。
guid	String	资产guid。
type_name	String	类型名称。
tags	Array of <b>TagHeader</b> objects	标签列表。

参数	参数类型	描述
classification_names	Array of strings	分类名称列表。

表 10-69 Connection

参数	参数类型	描述
guid	String	关联guid。
display_text	String	显示内容。
type_name	String	类型名称。
connection_type	String	连接类型。
qualified_name	String	限定名称。

表 10-70 TagHeader

参数	参数类型	描述
name	String	资产名称。
dexcription	Object	标签描述。
display_text	String	标签的名称。
relation_guid	String	关联的guid。
tag_guid	String	标签关联的guid。

状态码： 400

表 10-71 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-72 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-73 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-74 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-75 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{  
  "search_all_attributes" : true,  
  "offset" : 0,  
  "limit" : 10,  
  "type_names" : [ "Database" ],  
  "query" : "cbu_training"  
}
```



## 响应示例

状态码： 200

OK。

```
{
  "count" : 1,
  "entities" : [ {
    "attributes" : {
      "owner" : "wangyongjie",
      "create_time" : null,
      "qualified_name" : "cbu_training@dli.0833a5737480d53b2f25c010dc1a7b88-workspace-61aa10df45e54431a1901cb3527adab8",
      "name" : "cbu_training",
      "description" : "",
      "security_level" : null
    },
    "classification_names" : [ ],
    "connection" : {
      "connection_type" : "dli",
      "display_text" : "dli",
      "guid" : "0fff4057-c206-4dc3-a4ac-73ffc332bc9a",
      "qualified_name" : "dli@0833a5737480d53b2f25c010dc1a7b88-workspace-61aa10df45e54431a1901cb3527adab8",
      "type_name" : "Connection"
    },
    "display_text" : "cbu_training",
    "guid" : "e2b12c35-48ee-441f-a357-e338a85f5d00",
    "tags" : [ ],
    "type_name" : "dli_database"
  } ],
  "scroll_id" : null
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowEntitiesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
```

```
        .withAk(ak)
        .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowEntitiesRequest request = new ShowEntitiesRequest();
OpenEntitySearchRequest body = new OpenEntitySearchRequest();
request.withBody(body);
try {
    ShowEntitiesResponse response = client.showEntities(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowEntitiesRequest()
        request.body = OpenEntitySearchRequest(
        )
        response = client.show_entities(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowEntitiesRequest{}
    request.Body = &model.OpenEntitySearchRequest{
    }
    response, err := client.ShowEntities(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.4.2 根据 guid 获取资产详情

### 功能介绍

根据表guid可以获得表的详情信息，表的详情信息包含column的信息，也可以根据column的guid直接获取column的信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/asset/entities/{guid}

表 10-76 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产的guid，获取方法请参见 <a href="#">数据开发作业ID</a> 。

### 请求参数

表 10-77 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

### 响应参数

状态码： 200

表 10-78 响应 Body 参数

参数	参数类型	描述
entity	<b>entity</b> object	资产详情实体。
referred_entities	Object	引用实体Map(String, OpenEntity), key: 资产guid, value: 资产信息。

表 10-79 entity

参数	参数类型	描述
attributes	Object	属性Map(String, Object), key: 属性名称, value: 属性值。
connection	<b>Connection</b> object	数据连接。
create_time	Number	创建时间。
created_by	String	创建人。
display_text	String	资产的名称。
guid	String	资产guid。
relationship_attributes	Object	相关的属性 Map(String, Object), key: 关联关系名称, value: 关联关系信息。
type_name	String	资产类型。
updated_by	String	更新人。
update_time	Number	更新时间。
tags	Array of <b>TagHeader</b> objects	标签信息列表。
classification_names	Array of strings	分类名称列表。

表 10-80 Connection

参数	参数类型	描述
guid	String	关联guid。
display_text	String	显示内容。
type_name	String	类型名称。

参数	参数类型	描述
connection_type	String	连接类型。
qualified_name	String	限定名称。

表 10-81 TagHeader

参数	参数类型	描述
name	String	资产名称。
description	Object	标签描述。
display_text	String	标签的名称。
relation_guid	String	关联的guid。
tag_guid	String	标签关联的guid。

状态码： 400

表 10-82 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-83 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-84 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-85 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

ok。

```
{
  "entity": {
    "attributes": {
      "code": "d96c8f4e-7f59-4313-bed2-b68ef60e31b9",
      "standardClassify": null,
      "description": "",
      "rule": null,
      "type": "STRING",
      "standardCode": null,
      "securityLevel": null,
      "propertyType": "DIM",
      "alias": "",
      "definition": null,
      "workspaceId": "61aa10df45e54431a1901cb3527adab8",
      "owner": "",
      "synonyms": null,
      "length": 0,
      "updateTime": 1661908856022,
      "dataMaintainOwner": null,
      "hasValueList": false,
      "nameEng": "name",
      "linkedStandardType": null,
      "selfDefinedFields": null,
      "dataMonitorOwner": null,
      "qualified_name":
"subject_area_group.subject_area.object.dli_table.name@Business.0833a5737480d53b2f25c010dc1a7b88-
workspace-61aa10df45e54431a1901cb3527adab8",
      "createTime": 1661908856022,
      "name": "姓名",
      "ruleOwner": null,
      "valueExample": null,
    }
  }
}
```

```
"valueScope" : null,
"parameters" : null
},
"classification_names" : null,
"connection" : null,
"create_time" : 1661908450270,
"created_by" : "username",
"display_text" : "姓名",
"guid" : "d05187d0-adb0-4953-8239-eedb88c21b30",
"relationship_attributes" : {
  "tableColumn" : {
    "display_text" : null,
    "guid" : "6e02fcf2-7e66-4a32-bded-1d98790a4397",
    "relationship_attributes" : null,
    "relationship_guid" : null,
    "type_name" : null
  },
  "entity" : {
    "display_text" : null,
    "guid" : "a970c4fb-ac97-4339-95e6-944912c58a2b",
    "relationship_attributes" : null,
    "relationship_guid" : null,
    "type_name" : null
  },
  "tags" : [ ]
},
"tags" : [ ],
"type_name" : "BusinessLogicEntityColumn",
"update_time" : 1661908450270,
"updated_by" : "username"
},
"referred_entities" : { }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowEntityInfoByGuidSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```



```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowEntityInfoByGuidRequest request = new ShowEntityInfoByGuidRequest();
request.withGuid("{guid}");
try {
    ShowEntityInfoByGuidResponse response = client.showEntityInfoByGuid(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowEntityInfoByGuidRequest()
        request.guid = "{guid}"
        response = client.show_entity_info_by_guid(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```

dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowEntityInfoByGuidRequest{}
    request.Guid = "{guid}"
    response, err := client.ShowEntityInfoByGuid(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	ok。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

### 10.4.3 添加或修改资产

#### 功能介绍

添加或修改资产。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/{project\_id}/asset

表 10-86 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-87 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-88 请求 Body 参数

参数	是否必选	参数类型	描述
entity	是	<a href="#">AtlasAssetEntity</a> object	资产信息。
referred_entities	否	Object	关联资产map Map(String, AtlasAssetEntity)，key: guid, value: 资产信息。

表 10-89 AtlasAssetEntity

参数	是否必选	参数类型	描述
type_name	是	String	类型名称。
guid	否	String	资产guid。
version	否	Integer	版本。

参数	是否必选	参数类型	描述
update_time	否	Number	修改时间。
update_user	否	String	修改人。
create_time	否	Number	创建时间。
create_user	否	String	创建人。
display_text	否	String	展示。
status	否	String	状态。
classifications	否	Array of <b>AtlasClassificationInfo</b> objects	分类信息。
meanings	否	Array of <b>TermAssignmentHeader</b> objects	关联任务。
relation_ship_attributes	否	Object	实体map Map(String, Object), key: 关联关系类型: value: 关联关系信息。
attributes	是	Object	实体map Map(String, Object), key: 属性名称, value: 属性值。

表 10-90 AtlasClassificationInfo

参数	是否必选	参数类型	描述
entity_guid	否	String	资产guid。
propagate	否	Boolean	是否传播。
validity_periods	否	Array of <b>TimeBoundary</b> objects	时间信息。
type_name	否	String	类型名称。
attributes	否	Object	实体map Map(String, Object), key: 属性名称, value: 属性值。

表 10-91 TimeBoundary

参数	是否必选	参数类型	描述
start_time	否	String	开始时间。
end_time	否	String	结束时间。
time_zone	否	String	时区。

表 10-92 TermAssignmentHeader

参数	是否必选	参数类型	描述
confidence	否	Integer	信任id。
steward	否	String	管理员。
source	否	String	来源。
status	否	String	状态 枚举值：DISCOVERED、PROPOSED、IMPORTED、VALIDATED、DEPRECATED、OBSOLETE、OTHER。
create_user	否	String	创建人。
expression	否	String	表达式。
display_text	否	String	展示信息。
term_guid	否	String	标签guid。
relation_guid	否	String	关联guid。

## 响应参数

状态码： 400

表 10-93 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-94 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-95 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-96 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-97 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "entity": {
    "attributes": {
      "owner": null,
      "comments": null,
      "qualifiedName": "postgres.dm_autotest.duoallfuhe013.dqc_create_timestamp@dws.dws-4autotest-nomodify.dws.myhuaweicloud.com-dws_4autotest_nomodify.0833a5737480d53b2f25c010dc1a7b88-workspace-1b59d3c777ad4d619b89eeac4f3cce87",
      "isPartitionColumn": false,
      "description": null,
    }
  }
}
```

```

"isPrimaryKey" : false,
"type" : "bigint",
"ordinalPosition" : 6,
"connectionType" : "dws",
"securityLevel" : null,
"connectionQName" : "dws@dws-4autotest-nomodify.dws.myhuaweicloud.com-
dws_4autotest_nomodify.0833a5737480d53b2f25c010dc1a7b88-
workspace-1b59d3c777ad4d619b89eeac4f3cce87",
"isNullable" : "true",
"name" : "dqc_create_timestamp",
"connectionId" : "null:8a94806e79e693a30179e972c4aa000c",
"alias" : null,
"table" : {
  "uniqueAttributes" : {
    "qualifiedName" : "postgres.dm_autotest.duoallfuhe013@dws.dws-4autotest-
nomodify.dws.myhuaweicloud.com-dws_4autotest_nomodify.0833a5737480d53b2f25c010dc1a7b88-
workspace-1b59d3c777ad4d619b89eeac4f3cce87"
  },
  "typeName" : "dws_table",
  "name" : "duoallfuhe013",
  "guid" : "bc9af691-3401-4a7e-959f-413359dafeb6"
}
},
"classifications" : null,
"create_time" : 1662567796444,
"create_user" : "user_demo",
"display_text" : "dqc_create_timestamp",
"guid" : "266b1194-1713-47c9-94be-fdac82023f2f",
"meanings" : null,
"relation_ship_attributes" : {
  "inputToProcesses" : [ ],
  "meanings" : [ ],
  "table" : {
    "relationshipAttributes" : {
      "typeName" : "dws_table_column"
    },
    "displayText" : "duoallfuhe013",
    "relationshipGuid" : "f78811cb-0dd8-4459-8fda-54ca44d66005",
    "typeName" : "dws_table",
    "guid" : "bc9af691-3401-4a7e-959f-413359dafeb6",
    "relationshipStatus" : "ACTIVE"
  },
  "outputFromProcesses" : [ ]
},
"status" : "ACTIVE",
"type_name" : "dws_column",
"update_time" : 1662567796444,
"update_user" : "user_demo",
"version" : 0
},
"referred_entities" : {
  "dws@dws-4autotest-nomodify.dws.myhuaweicloud.com-
dws_4autotest_nomodify.0833a5737480d53b2f25c010dc1a7b88-
workspace-1b59d3c777ad4d619b89eeac4f3cce87" : {
    "attributes" : {
      "owner" : "user_demo",
      "securityLevel" : null,
      "createTime" : 0,
      "port" : 8000,
      "qualifiedName" : "dws@dws-4autotest-nomodify.dws.myhuaweicloud.com-
dws_4autotest_nomodify.0833a5737480d53b2f25c010dc1a7b88-
workspace-1b59d3c777ad4d619b89eeac4f3cce87",
      "name" : "dws_test",
      "host" : null,
      "description" : null,
      "id" : "8a94806e79e693a30179e972c4aa000c",
      "connectionType" : "dws"
    },
    "classifications" : null,

```

```
"create_time": 1665641601524,
"create_user": "user_demo",
"display_text": "dws_test",
"guid": "46bdc502-7912-460f-9d67-15141e9ab096",
"meanings": null,
"relation_ship_attributes": {
  "meanings": [ ]
},
"status": "ACTIVE",
"type_name": "Connection",
"update_time": 1669914264150,
"update_user": "user_demo",
"version": 0
}
}
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateOrUpdateAssetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateOrUpdateAssetRequest request = new CreateOrUpdateAssetRequest();
        EntityWithExtInfo body = new EntityWithExtInfo();
        request.withBody(body);
        try {
            CreateOrUpdateAssetResponse response = client.createOrUpdateAsset(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```



```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateOrUpdateAssetRequest()
        request.body = EntityWithExtInfo(
        )
        response = client.create_or_update_asset(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateOrUpdateAssetRequest{
    request.Body = &model.EntityWithExtInfo{
    }
}
response, err := client.CreateOrUpdateAsset(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.4.4 删除资产

### 功能介绍

删除资产。

### 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/{project\_id}/asset/{guid}

表 10-98 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

## 请求参数

表 10-99 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 400

表 10-100 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-101 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

状态码： 403

表 10-102 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-103 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-104 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteAssetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAssetRequest request = new DeleteAssetRequest();
        request.withGuid("{guid}");
        try {
            DeleteAssetResponse response = client.deleteAsset(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteAssetRequest()
    request.guid = "{guid}"
    response = client.delete_asset(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAssetRequest{}
    request.Guid = "{guid}"
    response, err := client.DeleteAsset(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.5 资产分类接口

### 10.5.1 批量资产关联分类

#### 功能介绍

批量资产关联分类：只支持对数据表的列和OBS对象添加分类。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/asset/entities/classification

表 10-105 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-106 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-107 请求 Body 参数

参数	是否必选	参数类型	描述
guids	是	Array of strings	数据资产list。
classification	是	<a href="#">OpenClassification</a> object	分类信息。

表 10-108 OpenClassification

参数	是否必选	参数类型	描述
name	是	String	分类名称。
description	否	String	分类描述。
create_user	否	String	分类创建者。
create_time	否	Number	分类创建时间。
update_time	否	Number	分类更新时间。
update_user	否	String	分类更新者。
guid	否	String	分类的guid标志。

## 响应参数

状态码： 400



表 10-109 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-110 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-111 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-112 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{  
  "guids": [ "0aa90b11-5064-4c6a-9c00-a35267a85aab", "348f0dc5-e03f-487e-a967-3e1abaccce8" ],  
  "classification": {  
    "name": "hive_common"  
  }  
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class BatchAssociateClassificationToEntitiesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchAssociateClassificationToEntitiesRequest request = new
        BatchAssociateClassificationToEntitiesRequest();
        OpenBulkClassifications body = new OpenBulkClassifications();
        request.withBody(body);
        try {
            BatchAssociateClassificationToEntitiesResponse response =
            client.batchAssociateClassificationToEntities(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchAssociateClassificationToEntitiesRequest()
        request.body = OpenBulkClassifications(
        )
        response = client.batch_associate_classification_to_entities(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchAssociateClassificationToEntitiesRequest{}
    request.Body = &model.OpenBulkClassifications{
    }
}
```

```
response, err := client.BatchAssociateClassificationToEntities(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.5.2 资产关联分类

### 功能介绍

将一个分类关联到一个或多个指定guid的资产上。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/asset/entities/guid/{guid}/classification

表 10-113 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

## 请求参数

表 10-114 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-115 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	分类名称。
description	否	String	分类描述。
create_user	否	String	分类创建者。
create_time	否	Number	分类创建时间。
update_time	否	Number	分类更新时间。
update_user	否	String	分类更新者。
guid	否	String	分类的guid标志。

## 响应参数

状态码： 400

表 10-116 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-117 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-118 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-119 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{  
  "name": "个人敏感数据"  
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class AssociateClassificationToEntitySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        AssociateClassificationToEntityRequest request = new AssociateClassificationToEntityRequest();
        request.withGuid("{guid}");
        OpenClassification body = new OpenClassification();
        request.withBody(body);
        try {
            AssociateClassificationToEntityResponse response = client.associateClassificationToEntity(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = AssociateClassificationToEntityRequest()  
    request.guid = "{guid}"  
    request.body = OpenClassification(  
    )  
    response = client.associate_classification_to_entity(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dataartsstudio.NewDataArtsStudioClient(  
        dataartsstudio.DataArtsStudioClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.AssociateClassificationToEntityRequest{}  
    request.Guid = "{guid}"  
    request.Body = &model.OpenClassification{  
    }  
    response, err := client.AssociateClassificationToEntity(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。



## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.5.3 移除资产关联的分类

### 功能介绍

移除资产关联分类。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/asset/entities/guid/{guid}/classification

表 10-120 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

### 请求参数

表 10-121 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-122 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	分类名称。
description	否	String	分类描述。
create_user	否	String	分类创建者。
create_time	否	Number	分类创建时间。
update_time	否	Number	分类更新时间。
update_user	否	String	分类更新者。
guid	否	String	分类的guid标志。

## 响应参数

状态码： 400

表 10-123 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-124 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-125 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-126 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

### 请求示例

```
{  
  "name": "个人敏感数据"  
}
```

### 响应示例

无

### 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.6 资产分级接口

### 10.6.1 批量资产关联密级

#### 功能介绍

批量资产关联密级：单个密级关联到多个资产上。

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v3/{project\_id}/asset/entities/security-level

表 10-127 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-128 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-129 请求 Body 参数

参数	是否必选	参数类型	描述
guids	是	Array of strings	资产guid。
security_level	是	String	密级。

## 响应参数

状态码： 400

表 10-130 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

状态码： 401

表 10-131 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-132 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-133 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{  
  "guids" : [ "a0a555e8-0e52-48ef-a067-bd0a7e255a29", "6a9862e2-62d3-481c-8486-a4c609bca740" ],  
  "security_level" : "l1"  
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class BatchAssociateSecurityLevelToEntitiesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchAssociateSecurityLevelToEntitiesRequest request = new
        BatchAssociateSecurityLevelToEntitiesRequest();
        BulkSecurityLevel body = new BulkSecurityLevel();
        request.withBody(body);
        try {
            BatchAssociateSecurityLevelToEntitiesResponse response =
            client.batchAssociateSecurityLevelToEntities(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = BatchAssociateSecurityLevelToEntitiesRequest()
    request.body = BulkSecurityLevel(
    )
    response = client.batch_associate_security_level_to_entities(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchAssociateSecurityLevelToEntitiesRequest{}
    request.Body = &model.BulkSecurityLevel{
    }
    response, err := client.BatchAssociateSecurityLevelToEntities(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.6.2 资产关联密级

### 功能介绍

关联资产到密级，资产关联指定密级。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/asset/entities/guid/{guid}/security-level

表 10-134 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。



表 10-135 Query 参数

参数	是否必选	参数类型	描述
security-level	是	String	资产密级。

## 请求参数

表 10-136 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 400

表 10-137 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-138 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-139 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-140 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class AssociateSecurityLevelToEntitieSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
AssociateSecurityLevelToEntitieRequest request = new AssociateSecurityLevelToEntitieRequest();
request.withGuid("{guid}");
try {
    AssociateSecurityLevelToEntitieResponse response = client.associateSecurityLevelToEntitie(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = AssociateSecurityLevelToEntitieRequest()
        request.guid = "{guid}"
        response = client.associate_security_level_to_entitie(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AssociateSecurityLevelToEntitieRequest{}
    request.Guid = "{guid}"
    response, err := client.AssociateSecurityLevelToEntitie(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.6.3 移除资产关联密级

### 功能介绍

移除资产关联密级。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{project\_id}/asset/entities/guid/{guid}/security-level

表 10-141 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

### 请求参数

表 10-142 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

### 响应参数

状态码： 400

表 10-143 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**状态码： 401**

**表 10-144** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**状态码： 403**

**表 10-145** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**状态码： 404**

**表 10-146** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteSecurityLevelFromEntitySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteSecurityLevelFromEntityRequest request = new DeleteSecurityLevelFromEntityRequest();
        request.withGuid("{guid}");
        try {
            DeleteSecurityLevelFromEntityResponse response = client.deleteSecurityLevelFromEntity(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = DeleteSecurityLevelFromEntityRequest()
    request.guid = "{guid}"
    response = client.delete_security_level_from_entity(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSecurityLevelFromEntityRequest{}
    request.Guid = "{guid}"
    response, err := client.DeleteSecurityLevelFromEntity(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。



## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.7 资产信息

### 10.7.1 查询概要

#### 功能介绍

查询概要。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{project\_id}/asset/profile

表 10-147 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 10-148 Query 参数

参数	是否必选	参数类型	描述
dw_id	是	String	数据连接ID。
db_type	是	String	数据库类型。
database_name	是	String	数据库名称。
table_name	是	String	表名。

## 请求参数

表 10-149 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-150 响应 Body 参数

参数	参数类型	描述
data	<a href="#">ProfileInfo</a> object	概要。
rowkey	String	行键。
status	String	状态。

表 10-151 ProfileInfo

参数	参数类型	描述
db_type	String	数据库类型。
cancel	Boolean	是否取消。
table_size	Number	表大小。
database_name	String	数据库名。
obs_common_config	String	obs公共配置。
total_row_count	String	总行数。
fields_name	Array of strings	文件列表。
table_name	String	表名。

参数	参数类型	描述
sample	String	样本。
update_date	String	修改时间。
row_count	Number	采样行数。
column_count	Number	列数。
unique	Boolean	是否唯一。
auto_stop	Boolean	自动停止。
time_profile	Boolean	时间档案
queue	String	队列。
dw_id	String	连接id。
columns_metric	Object	列概要信息。
columns_list	Array of <b>columnInfo</b> objects	列信息。

表 10-152 columnInfo

参数	参数类型	描述
column_name	String	列名称。
description	String	描述。
type	String	类型。

状态码： 400

表 10-153 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-154 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-155 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-156 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
{
  "status": "success",
  "rowkey": "1668584342946",
  "data": {
    "db_type": "DWS",
    "cancel": false,
    "table_size": 24576,
    "database_name": "wk",
    "obs_common_config": null,
    "total_row_count": 1,
    "fields_name": [],
    "columns_metric": {
      "num": {
        "min": null,
        "avg": null,
        "top": [ [ "1", null, 1 ] ],

```

```
"null" : 1,
"max" : null,
"variance" : null,
"topcount" : 1,
"stddev_samp" : null,
"distinct" : "0",
"duplicate" : 0,
"nonNull" : "1"
},
"sno" : {
"avglength" : 4,
"distinct" : "1",
"duplicate" : 0,
"maxlength" : 4,
"minlength" : 4,
"nonNull" : "1",
"null" : 0,
"top" : [ [ "1", "adff", 1 ] ],
"topcount" : 1
}
},
"columns_list" : [ {
"column_name" : "sno",
"description" : "",
"type" : "character varying"
}, {
"column_name" : "bon1",
"description" : "",
"type" : "integer"
}
]
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowDataProfileSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowDataProfileRequest request = new ShowDataProfileRequest();
try {
    ShowDataProfileResponse response = client.showDataProfile(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowDataProfileRequest()
        response = client.show_data_profile(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)
```

```

)
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDataProfileRequest{}
    response, err := client.ShowDataProfile(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.7.2 指定字段采集概要

### 功能介绍

指定字段采集概要信息接口。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/{project\_id}/asset/profile/columns

表 10-157 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-158 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-159 请求 Body 参数

参数	是否必选	参数类型	描述
dw_id	否	String	连接id。
db_type	否	String	数据源类型。
database_name	否	String	数据库名称。
table_name	否	String	表名称。
table_id	否	String	表ID。
column_names	否	Array of <a href="#">TableColumn DTO</a> objects	待更新概要字段列表。
fields_name	否	Array of strings	字段名称列表。
queue	否	String	执行更新语句队列。



参数	是否必选	参数类型	描述
unique	否	Boolean	是否采集唯一值。
time_profile	否	Boolean	时间档案。
sample	否	String	数据采样方式。
job_id	否	String	作业id。
cancel	否	Boolean	是否取消。
auto_stop	否	Boolean	是否自动停止。
obsconfig	否	<b>OBSCCommon Config</b> object	obs配置，概要采集不涉及。

表 10-160 TableColumnDTO

参数	是否必选	参数类型	描述
column_name	否	String	字段名称。
description	否	String	字段描述。
type	否	String	字段类型。
is_partition_column	否	Boolean	是否是分区字段。

表 10-161 OBSCCommonConfig

参数	是否必选	参数类型	描述
column_map	否	Object	Map<String, String>结构，key: 字段名称，value: 字段信息
path	否	String	路径。
delimiter	否	String	分隔符。
quote	否	String	引用。
escape	否	String	规避。
header	否	Boolean	是否是标头。
data_type	否	String	数据类型。
date_format	否	String	数据格式。
timestamp_format	否	String	时间格式。

参数	是否必选	参数类型	描述
null_value	否	String	为空时默认值。
comment	否	String	注解。
parse_mode	否	String	解析模式。
join_table	否	String	联表。

## 响应参数

状态码： 400

表 10-162 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-163 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-164 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-165 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "dw_id": "003b3ed52daf41e6829a0bc74526f5f7",
  "db_type": "DLI",
  "database_name": "default",
  "table_name": "a1",
  "queue": "default",
  "unique": false,
  "sample": "0",
  "column_names": [ {
    "column_name": "q1",
    "description": "",
    "type": "double"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class RenewDataProfileSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
RenewDataProfileRequest request = new RenewDataProfileRequest();
DataProfileRO body = new DataProfileRO();
request.withBody(body);
try {
    RenewDataProfileResponse response = client.renewDataProfile(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RenewDataProfileRequest()
        request.body = DataProfileRO(
        )
        response = client.renew_data_profile(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
```

```

"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RenewDataProfileRequest{}
    request.Body = &model.DataProfileRo{
    }
    response, err := client.RenewDataProfile(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.8 血缘信息

## 10.8.1 无血缘关系表查询

### 功能介绍

无血缘关系表查询，请求参数type\_names、classification、type\_names、connection\_names、query需至少指定一个。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/lineage/search/unrelated/table

表 10-166 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 10-167 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-168 请求 Body 参数

参数	是否必选	参数类型	描述
attributes	否	Object	属性。
classifications	否	Array of strings	分类。
connection_names	否	Array of strings	数据连接。
exclude_classifications	否	Boolean	是否排除分类。

参数	是否必选	参数类型	描述
exclude_security_levels	否	Boolean	是否排除密级。
exclude_tags	否	Boolean	是否排除标签。
include_classification_attributes	否	Boolean	包含分类属性。
include_sub_classifications	否	Boolean	包含子分类。
limit	否	Integer	分页参数，每页限制数量，默认查询所有。
offset	否	Integer	分页参数，偏移量，默认查询所有。
order	否	String	排序方式。
query	否	String	查询参数。
search_name_and_description	否	Boolean	是否按名称和描述搜索。
security_levels	否	Array of strings	安全密级列表。
term_names	否	Array of strings	标签列表。
type_names	否	Array of strings	类型列表。

## 响应参数

状态码： 200

表 10-169 响应 Body 参数

参数	参数类型	描述
attributes	<b>AttributeSearchResult</b> object	属性信息。
classification	String	分类。
count	Integer	结果总量。

参数	参数类型	描述
entities	Array of <a href="#">AtlasEntityHeader</a> objects	资产信息。
full_text_result	Array of <a href="#">AtlasFullTextResult</a> objects	全文结果。
query_text	String	查询内容。
query_type	String	查询类型，取值范围DSL、FULL_TEXT、GREMLIN、BASIC、ATTRIBUTE、RELATIONSHIP、ADVANCED。
referred_entities	Object	Map(String, AtlasEntityHeader), key: guid, value: 资产信息AtlasEntityHeader。
scroll_id	String	滚动条id。
search_parameters	Object	参数。
type	String	类型。

表 10-170 AttributeSearchResult

参数	参数类型	描述
name	Array of strings	名称列表。
values	Array of objects	值列表。

表 10-171 AtlasEntityHeader

参数	参数类型	描述
type_name	String	类型名称。
guid	String	资产guid。
name	String	名称。
display_text	String	展示。
status	String	状态。
classification_names	Array of strings	分类名称列表。



参数	参数类型	描述
classifications	Array of <b>AtlasClassificationInfo</b> objects	分类信息。
meaning_names	Array of strings	标签名称列表。
meanings	Array of <b>TermAssignmentHeader</b> objects	关联任务。
children	Object	实体map Map(String, AtlasEntityHeader)。key: guid, value: 资产信息。
attributes	Object	实体map Map(String, Object)。key: 属性名称, value: 属性值。

表 10-172 AtlasFullTextResult

参数	参数类型	描述
entity	<b>AtlasEntityHeader</b> object	资产信息。
score	Number	数值。

表 10-173 AtlasClassificationInfo

参数	参数类型	描述
entity_guid	String	资产guid。
propagate	Boolean	是否传播。
validity_periods	Array of <b>TimeBoundary</b> objects	时间信息。
type_name	String	类型名称。
attributes	Object	实体map Map(String, Object), key: 属性名称, value: 属性值。

表 10-174 TimeBoundary

参数	参数类型	描述
start_time	String	开始时间。
end_time	String	结束时间。
time_zone	String	时区。

表 10-175 TermAssignmentHeader

参数	参数类型	描述
confidence	Integer	信任id。
steward	String	管理员。
source	String	来源。
status	String	状态 枚举值: DISCOVERED、PROPOSED、IMPORTED、VALIDATED、DEPRECATED、OBSOLETE、OTHER。
create_user	String	创建人。
expression	String	表达式。
display_text	String	展示信息。
term_guid	String	标签guid。
relation_guid	String	关联guid。

状态码: 400

表 10-176 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码: 401

表 10-177 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

状态码： 403

表 10-178 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-179 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "attributes": {
    "database": [ "database" ],
    "classifications": [ ],
    "connection_names": [ ],
    "exclude_classifications": false,
    "exclude_security_levels": false,
    "exclude_tags": false,
    "include_classification_attributes": false,
    "include_sub_classifications": false,
    "limit": 10,
    "offset": 0,
    "order": "incr",
    "query": "wk_02",
    "search_name_and_description": true,
    "security_levels": [ ],
    "term_names": [ ],
    "type_names": [ ]
  }
}
```

## 响应示例

状态码： 200

OK。

```
{
  "attributes": null,
}
```

```
"classification" : null,
"count" : 78,
"entities" : [ {
  "attributes" : {
    "securityLevel" : null,
    "owner" : "DA_test",
    "comments" : null,
    "createTime" : 1674961143000,
    "qualifiedName" : "wk.hive_02@hive.192.168.0.96-
mrs_3x_autotest_do_not_del.0833a5737480d53b2f25c010dc1a7b88-workspace -
b88c445407b24283aa949f9833a38fd8",
    "connectionQName" : "hive@192.168.0.96-
mrs_3x_autotest_do_not_del.0833a5737480d53b2f25c010dc1a7b88-workspace -
b88c445407b24283aa949f9833a38fd8",
    "name" : "hive_02",
    "description" : null,
    "connectionType" : "hive"
  },
  "children" : null,
  "classification_names" : [ ],
  "classifications" : null,
  "display_text" : null,
  "guid" : "9915ee1e-3803-4f4d-86b2-ee5dfadc0d46",
  "meaning_names" : [ ],
  "meanings" : [ ],
  "name" : null,
  "status" : "ACTIVE",
  "type_name" : "hive_table"
} ],
"full_text_result" : null,
"query_text" : "",
"query_type" : "ADVANCED",
"referred_entities" : null,
"scroll_id" : null,
"search_parameters" : null,
"type" : null
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowUnrelatedTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
    }
}
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowUnrelatedTableRequest request = new ShowUnrelatedTableRequest();
SearchParametersExt body = new SearchParametersExt();
request.withBody(body);
try {
    ShowUnrelatedTableResponse response = client.showUnrelatedTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowUnrelatedTableRequest()
        request.body = SearchParametersExt(
        )
        response = client.show_unrelated_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowUnrelatedTableRequest{}
    request.Body = &model.SearchParametersExt{
    }
    response, err := client.ShowUnrelatedTable(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.8.2 血缘查询

### 功能介绍

血缘查询。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/entities/{guid}/lineage

表 10-180 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

表 10-181 Query 参数

参数	是否必选	参数类型	描述
direction	否	String	查询方向，取值范围：BOTH、IN、OUT。默认BOTH。
depth	否	Integer	血缘链路长度，默认值5。

### 请求参数

表 10-182 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-183 响应 Body 参数

参数	参数类型	描述
base_entity_guid	String	当前资产的guid。
guid_entity_map	Object	实体集合Map(String, OpenEntityHeader)。key: 资产guid, value: 资产信息OpenEntityHeader。
relations	Array of <a href="#">LineageRelation</a> objects	血缘关系。
referred_entities	Object	相关实体集合Map(String, OpenEntity)。key: 资产guid, value: 资产信息OpenEntity。

表 10-184 LineageRelation

参数	参数类型	描述
from_entity_id	String	血缘来源资产guid。
relationship_id	String	关系id。
to_entity_id	String	血缘流向资产guid。

状态码： 400

表 10-185 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401



表 10-186 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-187 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-188 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
{
  "base_entity_guid": "c1cba918-24d3-4dc4-a65c-178e57b14458",
  "guid_entity_map": {
    "c1cba918-24d3-4dc4-a65c-178e57b14458": {
      "attributes": {
        "owner": "userName",
        "create_time": 0,
        "qualified_name": "MRS_Flink_Job_001.flink_01@FLINK.f9a70740-6a5e-492a-a369-9779fd952f85.0833a5737480d53b2f25c010dc1a7b88-workspace - b88c445407b24283aa949f9833a38fd8",
        "name": "flink_01",
        "description": "This temp entity was generated from DLF lineage service",
        "security_level": null
      },
      "classification_names": [],
      "connection": null,
    }
  }
}
```

```

"display_text": "flink_01",
"guid": "c1cba918-24d3-4dc4-a65c-178e57b14458",
"tags": [],
"type_name": "flink_table"
},
"223d0579-cf6f-4439-950b-7372a0f1a5a2": {
  "attributes": {
    "owner": "userName",
    "create_time": 0,
    "qualified_name": "MRS_Flink_Job_001.flink_02@FLINK.f9a70740-6a5e-492a-
a369-9779fd952f85.0833a5737480d53b2f25c010dc1a7b88-workspace -
b88c445407b24283aa949f9833a38fd8",
    "name": "flink_02",
    "description": "This temp entity was generated from DLF lineage service",
    "security_level": null
  },
  "classification_names": [],
  "connection": null,
  "display_text": "flink_02",
  "guid": "223d0579-cf6f-4439-950b-7372a0f1a5a2",
  "tags": [],
  "type_name": "flink_table"
},
"e2f63e6a-206d-4090-a032-7e038d507269": {
  "attributes": {
    "owner": "userName",
    "outputs": [ {
      "uniqueAttributes": {
        "qualifiedName": "MRS_Flink_Job_001.flink_01@FLINK.f9a70740-6a5e-492a-
a369-9779fd952f85.0833a5737480d53b2f25c010dc1a7b88-workspace -
b88c445407b24283aa949f9833a38fd8"
      }
    }
  ],
  "typeName": "flink_table",
  "name": "flink_01",
  "guid": "c1cba918-24d3-4dc4-a65c-178e57b14458"
}],
"create_time": null,
"qualified_name": "509207.42419233-b3f5-42b3-
a314-336132c23328@dlf_job.0833a5737480d53b2f25c010dc1a7b88-workspace
-1b59d3c777ad4d619b89eeac4f3cce87",
"inputs": [ {
  "uniqueAttributes": {
    "qualifiedName": "MRS_Flink_Job_001.flink_02@FLINK.f9a70740-6a5e-492a-
a369-9779fd952f85.0833a5737480d53b2f25c010dc1a7b88-workspace -
b88c445407b24283aa949f9833a38fd8"
  }
}
],
"typeName": "flink_table",
"name": "flink_02",
"guid": "223d0579-cf6f-4439-950b-7372a0f1a5a2"
}],
"name": "MRS_Flink_Job_001",
"description": "Workspace: kw_test",
"security_level": "l1"
},
"classification_names": [],
"connection": null,
"display_text": "MRS_Flink_Job_001",
"guid": "e2f63e6a-206d-4090-a032-7e038d507269",
"tags": [],
"type_name": "flink_node"
}
},
"referred_entities": { },
"relations": [ {
  "from_entity_id": "e2f63e6a-206d-4090-a032-7e038d507269",
  "relationship_id": "b5ec94b0-e779-49a8-edef-f6bbf46bf63c",
  "to_entity_id": "c1cba918-24d3-4dc4-a65c-178e57b14458"
}
], {
  "from_entity_id": "223d0579-cf6f-4439-950b-7372a0f1a5a2",

```

```
"relationship_id" : "7a1b8777-3bd9-460a-b729-70d3d3bf3744",  
"to_entity_id" : "e2f63e6a-206d-4090-a032-7e038d507269"  
}]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowLineageSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowLineageRequest request = new ShowLineageRequest();  
        request.withGuid("{guid}");  
        try {  
            ShowLineageResponse response = client.showLineage(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowLineageRequest()
        request.guid = "{guid}"
        response = client.show_lineage(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowLineageRequest{}
```

```
request.Guid = "{guid}"
response, err := client.ShowLineage(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.8.3 血缘导入

### 功能介绍

血缘查询。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/lineage/import

表 10-189 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-190 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-191 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of <a href="#">TableLineage</a> objects	请求参数。

表 10-192 TableLineage

参数	是否必选	参数类型	描述
name	是	String	作业算子名称。
input_tables	是	Array of <a href="#">TableInfo</a> objects	上游血缘表列表，列表大小：1至100。
output_tables	是	Array of <a href="#">TableInfo</a> objects	下游血缘表列表，列表大小：1至100。
source_connection_id	是	String	源数据连接id。
target_connection_id	否	String	目标数据连接id。
column_lineages	否	Array of <a href="#">ColumnLineage</a> objects	字段血缘列表，列表大小：0至100。

表 10-193 TableInfo

参数	是否必选	参数类型	描述
database	否	String	数据库名称。
schema	否	String	逻辑库名称。
table	否	String	表名称。

表 10-194 ColumnLineage

参数	是否必选	参数类型	描述
name	是	String	作业算子名称。
input_columns	是	Array of <b>ColumnDetails</b> objects	上游血缘字段列表，列表大小：1至100。
output_columns	是	Array of <b>ColumnDetails</b> objects	下游血缘字段列表，列表大小：1至100。

表 10-195 ColumnDetails

参数	是否必选	参数类型	描述
database	否	String	数据库名称。
schema	否	String	逻辑库名称。
table	否	String	表名称。
column	否	String	字段名称。

## 响应参数

状态码： 200

表 10-196 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of <b>ObjectInfo</b> objects	血缘导入结果

表 10-197 ObjectIdInfo

参数	参数类型	描述
name	String	作业算子名称。
type_name	String	资产类型。
qualified_name	String	作业资产唯一限定名称。

状态码： 400

表 10-198 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-199 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-200 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404



表 10-201 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
[ {
  "name" : "test1221",
  "input_tables" : [ {
    "database" : "wk",
    "schema" : null,
    "table" : "wk_shuju1"
  } ],
  "output_tables" : [ {
    "database" : "wk",
    "schema" : null,
    "table" : "wk_shuju2"
  } ],
  "source_connection_id" : "aa0e89b9c7c14a6b9737d56a53d7a286",
  "target_connection_id" : "aa0e89b9c7c14a6b9737d56a53d7a286",
  "column_lineages" : [ ]
} ]
```

## 响应示例

状态码： 200

OK。

```
[ {
  "name" : "test",
  "type_name" : "Node",
  "qualified_name" : "manual.a0683065-cfb6-42d3-a0ff-87b2cc5e3c79@node.0833a5737480d53b2f25c010dc1a7b88-workspace-ee119e8faee347a389e8c295b926331c"
} ]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ImportLineageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ImportLineageRequest request = new ImportLineageRequest();
try {
    ImportLineageResponse response = client.importLineage(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ImportLineageRequest()
        response = client.import_lineage(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ImportLineageRequest{}
    response, err := client.ImportLineage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.8.4 创建血缘信息

### 功能介绍

创建血缘信息

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/lineage/lineage-info

表 10-202 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 10-203 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a>

表 10-204 请求 Body 参数

参数	是否必选	参数类型	描述
cluster_id	否	String	集群id
data_source_type	否	String	数据连接类型
connection_id	否	String	数据连接id

参数	是否必选	参数类型	描述
connection_name	否	String	数据连接名称
workspace_id	否	String	工作空间id
job_id	否	String	作业id
node_name	否	String	算子名称
table_lineage	否	<a href="#">TableLineageV2</a> object	血缘对象

表 10-205 TableLineageV2

参数	是否必选	参数类型	描述
input_tables	是	Array of <a href="#">TableInfoV2</a> objects	上游血缘表列表，列表大小：1至100
output_tables	是	Array of <a href="#">TableInfoV2</a> objects	下游血缘表列表，列表大小：1至100
column_lineages	否	Array of <a href="#">ColumnLineageV2</a> objects	字段血缘列表，列表大小：0至100

表 10-206 TableInfoV2

参数	是否必选	参数类型	描述
catalog	否	String	目录名称
database	否	String	数据库名称
schema	否	String	逻辑库名称
table	否	String	表名称

表 10-207 ColumnLineageV2

参数	是否必选	参数类型	描述
input_columns	是	Array of <a href="#">ColumnDetails</a> objects	上游血缘字段列表，列表大小：1至100

参数	是否必选	参数类型	描述
output_columns	是	Array of <b>ColumnDetails</b> objects	下游血缘字段列表，列表大小：1至100

表 10-208 ColumnDetails

参数	是否必选	参数类型	描述
database	否	String	数据库名称。
schema	否	String	逻辑库名称。
table	否	String	表名称。
column	否	String	字段名称。

## 响应参数

状态码： 200

表 10-209 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of <b>ObjectIdInfo</b> objects	血缘导入结果

表 10-210 ObjectIdInfo

参数	参数类型	描述
name	String	作业算子名称。
type_name	String	资产类型。
qualified_name	String	作业资产唯一限定名称。

状态码： 400

表 10-211 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-212 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-213 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-214 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{  
  "cluster_id": "actual_cluster_id",  
  "data_source_type": "actual_data_source_type",  
  "connection_id": "003b3ed52daf41e6829a0bc74526f5f7",  
  "connection_name": "dli_test",  
  "workspace_id": "1b59d3c777ad4d619b89eeac4f3cce87",  
  "job_id": "testJobID",  
  "node_name": "testNodeMName",  
  "table_lineage": {  
    "input_tables": [ {
```

```

"catalog": "spark_catalog",
"database": "nbbtemp",
"schema": "",
"table": "origndata_typeall"
}],
"output_tables": [ {
"catalog": "spark_catalog",
"database": "nbbtemp",
"schema": "",
"table": "lineageTable1"
}],
"column_lineages": [ {
"input_columns": [ {
"database": "nbbtemp",
"schema": "",
"table": "origndata_typeall",
"column": "stringf"
}],
"output_columns": [ {
"database": "nbbtemp",
"schema": "",
"table": "lineageTable1",
"column": "col1"
}]
}]
}
}
}

```

## 响应示例

**状态码: 200**

OK

```

[ {
" name": "test",
" type_name": "Node",
" qualified_name": "manual.a0683065-cfb6-42d3-
a0ff-87b2cc5e3c79@node.0833a5737480d53b2f25c010dc1a7b88-workspace-
ee119e8faee347a389e8c295b926331c"
} ]

```

## SDK 代码示例

SDK代码示例如下。

### Java

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateLineageInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
    }
}

```



this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
CreateLineageInfoRequest request = new CreateLineageInfoRequest();
LineageInfoRequest body = new LineageInfoRequest();
request.withBody(body);
try {
    CreateLineageInfoResponse response = client.createLineageInfo(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateLineageInfoRequest()
        request.body = LineageInfoRequest(
        )
        response = client.create_lineage_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateLineageInfoRequest{}
    request.Body = &model.LineageInfoRequest{}
    response, err := client.CreateLineageInfo(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	BadRequest
401	Unauthorized
403	Forbidden

状态码	描述
404	Not Found

## 10.9 元数据采集任务接口

### 10.9.1 查询采集任务列表

#### 功能介绍

查询采集任务列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/metadata/tasks/search

表 10-215 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 10-216 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-217 请求 Body 参数

参数	是否必选	参数类型	描述
user_name	否	String	创建人。
name	否	String	任务名。
data_source_type	否	String	数据源类型。
data_connection_id	否	String	数据连接id。
start_time	否	String	开始时间。
end_time	否	String	结束时间。
directory_id	否	String	目录id。
bucket_name	否	String	桶名。
limit	否	Integer	分页参数limit, 默认值: 10。
offset	否	Integer	分页参数offset, 默认值: 0。

## 响应参数

状态码: 200

表 10-218 响应 Body 参数

参数	参数类型	描述
count	Integer	查询采集任务数量。
total_records	Integer	同一projectId下已创建采集任务数量。
max_records	Integer	同一projectId下允许创建采集任务数量。
resources	Array of <a href="#">MetadataCollectionTask</a> objects	采集任务列表。

表 10-219 MetadataCollectionTask

参数	参数类型	描述
id	String	任务id。
name	String	任务名称。
description	String	任务描述。

参数	参数类型	描述
user_id	String	用户id。
create_time	String	创建时间。
project_id	String	产品id。
dir_id	String	目录id。
schedule_config	<b>SchedulerInfo</b> object	调度信息。
parameter_config	Array of <b>CustomMetadata</b> objects	自定义元数据信息。
update_time	String	修改时间。
user_name	String	用户名。
path	String	路径。
last_run_time	String	最后一次执行时间。
start_time	String	开始时间。
end_time	String	结束时间。
next_run_time	String	下一次执行时间。
duty_person	String	责任人。
update_type	String	修改类型。
data_source_type	String	数据来源类型。
task_config	Object	任务信息Map(String, Object)。key: 属性名称, value: 属性值。
data_source_workspace_id	String	数据来源工作空间id。

表 10-220 SchedulerInfo

参数	参数类型	描述
cron_expression	String	表达式。
end_time	String	结束时间。
max_time_out	Integer	最大超时时间。
interval	String	间隔。

参数	参数类型	描述
schedule_type	String	调度类型。
start_time	String	开始时间。
job_id	Integer	工作id。
enabled	Integer	启用。

表 10-221 CustomMetadata

参数	参数类型	描述
description	String	描述。
type	String	类型。
task_id	String	任务id。

状态码： 400

表 10-222 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-223 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-224 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

状态码： 404

表 10-225 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "user_name": "userName",
  "name": null,
  "data_source_type": null,
  "data_connection_id": null,
  "start_time": null,
  "end_time": null,
  "directory_id": "e4248d66-c376-4f1b-a576-e4d1c6838d83",
  "bucket_name": null,
  "limit": 10,
  "offset": 1
}
```

## 响应示例

状态码： 200

OK。

```
{
  "count": 1,
  "total_records": 1,
  "max_records": 100,
  "resources": [ {
    "create_time": "1663722819809",
    "data_source_type": "HIVE",
    "data_source_workspace_id": null,
    "description": null,
    "dir_id": "0",
    "duty_person": "user_demo",
    "end_time": "0",
    "id": "8a948047835b945101835d9b90e1010f",
    "last_run_time": "1663723446000",
    "name": "hive921",
    "next_run_time": "0",
    "parameter_config": [ ],
    "path": null,
    "project_id": null,
    "schedule_config": [ {
      "cron_expression": null,
      "enabled": 4,
      "end_time": null,
      "interval": null,

```

```

"job_id" : 262070,
"max_time_out" : 60,
"schedule_type" : "RUN_ONCE",
"start_time" : null
}],
"start_time" : "0",
"task_config" : [ ],
"update_time" : "1663722819809",
"update_type" : null,
"user_id" : "0833a573fe80d5401f6dc010a775569a",
"user_name" : "user_demo"
}]
}

```

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.9.2 创建采集任务

### 功能介绍

创建采集任务。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/metadata/tasks/create

表 10-226 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。



## 请求参数

表 10-227 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-228 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	任务id。
name	是	String	任务名称。
description	否	String	任务描述。
user_id	是	String	用户id。
create_time	否	String	创建时间。
project_id	否	String	产品id。
dir_id	是	String	目录id。
schedule_config	是	<a href="#">SchedulerInfo</a> object	调度信息。
parameter_config	否	Array of <a href="#">CustomMetadata</a> objects	自定义元数据信息。
update_time	否	String	修改时间。
user_name	否	String	用户名。
path	否	String	路径。
last_run_time	否	String	最后一次执行时间。
start_time	否	String	开始时间。
end_time	否	String	结束时间。
next_run_time	否	String	下一次执行时间。
duty_person	否	String	责任人。
update_type	否	String	修改类型。

参数	是否必选	参数类型	描述
data_source_type	是	String	数据来源类型。
task_config	是	Object	任务信息Map(String, Object)。key: 属性名称, value: 属性值。
data_source_workspace_id	否	String	数据来源工作空间id。

表 10-229 SchedulerInfo

参数	是否必选	参数类型	描述
cron_expression	否	String	表达式。
end_time	否	String	结束时间。
max_time_out	否	Integer	最大超时时间。
interval	否	String	间隔。
schedule_type	否	String	调度类型。
start_time	否	String	开始时间。
job_id	否	Integer	工作id。
enabled	否	Integer	启用。

表 10-230 CustomMetadata

参数	是否必选	参数类型	描述
description	否	String	描述。
type	否	String	类型。
task_id	否	String	任务id。

## 响应参数

状态码： 200

表 10-231 响应 Body 参数

参数	参数类型	描述
task_id	String	任务id。

状态码： 400

表 10-232 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-233 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-234 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-235 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "id" : null,
  "data_source_type" : "DLI",
  "description" : "",
  "name" : "test采集110",
  "user_id" : "0833a573fe80d5401f6dc010a775569a",
  "dir_id" : "0",
  "task_config" : {
    "data_connection_name" : "dli_test",
    "data_connection_id" : "14bb947454e64de784eadaab759081e8",
    "databaseName" : [ "ygy1" ],
    "tableName" : [ "ygy1.ygy" ],
    "alive_object_policy" : "3",
    "deleted_object_policy" : "3",
    "enableDataProfile" : false,
    "enableDataClassification" : false,
    "enable_table_filter_by_regex" : false
  },
  "schedule_config" : {
    "schedule_type" : "RUN_ONCE",
    "max_time_out" : 10
  }
}
```

## 响应示例

状态码： 200

OK。

```
{
  "task_id" : "8a9480ed8353e491018354b87f63009e"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
CreateTaskRequest request = new CreateTaskRequest();
MetadataCollectionTask body = new MetadataCollectionTask();
request.withBody(body);
try {
    CreateTaskResponse response = client.createTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateTaskRequest()
        request.body = MetadataCollectionTask(
        )
        response = client.create_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateTaskRequest{}
    request.Body = &model.MetadataCollectionTask{
    }
    response, err := client.CreateTask(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.9.3 查询采集任务详情

### 功能介绍

查询采集任务详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/metadata/tasks/{task\_id}

表 10-236 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
task_id	是	String	任务id。

### 请求参数

表 10-237 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

### 响应参数

状态码： 200

表 10-238 响应 Body 参数

参数	参数类型	描述
id	String	任务id。
name	String	任务名称。

参数	参数类型	描述
description	String	任务描述。
user_id	String	用户id。
create_time	String	创建时间。
project_id	String	产品id。
dir_id	String	目录id。
schedule_config	<b>SchedulerInfo</b> object	调度信息。
parameter_config	Array of <b>CustomMetadata</b> objects	自定义元数据信息。
update_time	String	修改时间。
user_name	String	用户名。
path	String	路径。
last_run_time	String	最后一次执行时间。
start_time	String	开始时间。
end_time	String	结束时间。
next_run_time	String	下一次执行时间。
duty_person	String	责任人。
update_type	String	修改类型。
data_source_type	String	数据来源类型。
task_config	Object	任务信息Map(String, Object)。key: 属性名称, value: 属性值。
data_source_workspace_id	String	数据来源工作空间id。

表 10-239 SchedulerInfo

参数	参数类型	描述
cron_expression	String	表达式。
end_time	String	结束时间。
max_time_out	Integer	最大超时时间。



参数	参数类型	描述
interval	String	间隔。
schedule_type	String	调度类型。
start_time	String	开始时间。
job_id	Integer	工作id。
enabled	Integer	启用。

表 10-240 CustomMetadata

参数	参数类型	描述
description	String	描述。
type	String	类型。
task_id	String	任务id。

状态码： 400

表 10-241 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-242 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-243 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-244 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
{
  "create_time": "1663573720930",
  "data_source_type": "HIVE",
  "data_source_workspace_id": null,
  "description": "",
  "dir_id": "0",
  "duty_person": "user_demo",
  "end_time": "0",
  "id": "8a9480ed8353e491018354b87f63009e",
  "last_run_time": "1663574072000",
  "name": "采集验证一汽大众需求",
  "next_run_time": "0",
  "parameter_config": [ ],
  "path": "",
  "project_id": null,
  "schedule_config": [ {
    "cron_expression": null,
    "enabled": 4,
    "end_time": null,
    "interval": null,
    "job_id": 261593,
    "max_time_out": 60,
    "schedule_type": "RUN_ONCE",
    "start_time": null
  } ],
  "start_time": "0",
  "task_config": [ ],
  "update_time": "1663573720930",
  "update_type": null,
  "user_id": "0833a573fe80d5401f6dc010a775569a",
}
```

```
"user_name" : "user_demo"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowTaskInfoSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowTaskInfoRequest request = new ShowTaskInfoRequest();  
        request.withTaskId("{task_id}");  
        try {  
            ShowTaskInfoResponse response = client.showTaskInfo(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowTaskInfoRequest()
        request.task_id = "{task_id}"
        response = client.show_task_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowTaskInfoRequest{}
    request.TaskId = "{task_id}"
    response, err := client.ShowTaskInfo(request)
```

```
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.9.4 编辑采集任务

### 功能介绍

编辑采集任务。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/metadata/tasks/{task\_id}

表 10-245 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
task_id	是	String	任务id。

## 请求参数

表 10-246 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-247 请求 Body 参数

参数	是否必选	参数类型	描述
id	否	String	任务id。
name	是	String	任务名称。
description	否	String	任务描述。
user_id	是	String	用户id。
create_time	否	String	创建时间。
project_id	否	String	产品id。
dir_id	是	String	目录id。
schedule_config	是	<a href="#">SchedulerInfo</a> object	调度信息。
parameter_config	否	Array of <a href="#">CustomMetadata</a> objects	自定义元数据信息。
update_time	否	String	修改时间。
user_name	否	String	用户名。
path	否	String	路径。
last_run_time	否	String	最后一次执行时间。
start_time	否	String	开始时间。
end_time	否	String	结束时间。
next_run_time	否	String	下一次执行时间。
duty_person	否	String	责任人。
update_type	否	String	修改类型。

参数	是否必选	参数类型	描述
data_source_type	是	String	数据来源类型。
task_config	是	Object	任务信息Map(String, Object)。key: 属性名称, value: 属性值。
data_source_workspace_id	否	String	数据来源工作空间id。

表 10-248 SchedulerInfo

参数	是否必选	参数类型	描述
cron_expression	否	String	表达式。
end_time	否	String	结束时间。
max_time_out	否	Integer	最大超时时间。
interval	否	String	间隔。
schedule_type	否	String	调度类型。
start_time	否	String	开始时间。
job_id	否	Integer	工作id。
enabled	否	Integer	启用。

表 10-249 CustomMetadata

参数	是否必选	参数类型	描述
description	否	String	描述。
type	否	String	类型。
task_id	否	String	任务id。

## 响应参数

状态码： 400

表 10-250 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-251 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-252 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-253 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{  
  "id": null,  
  "data_source_type": "DLI",  
  "description": "",  
  "name": "test采集110",  
  "user_id": "0833a573fe80d5401f6dc010a775569a",  
  "dir_id": "0",  
  "task_config": {  
    "data_connection_name": "dli_test",  
    "data_connection_id": "14bb947454e64de784eadaab759081e8",  
  }  
}
```



```
"databaseName" : [ "ygy1" ],
"tableName" : [ "ygy1.ygy" ],
"alive_object_policy" : "3",
"deleted_object_policy" : "3",
"enableDataProfile" : false,
"enableDataClassification" : false,
"enable_table_filter_by_regex" : false
},
"schedule_config" : {
"schedule_type" : "RUN_ONCE",
"max_time_out" : 10
}
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateTaskInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTaskInfoRequest request = new UpdateTaskInfoRequest();
        request.withTaskId("{task_id}");
        MetadataCollectionTask body = new MetadataCollectionTask();
        request.withBody(body);
        try {
            UpdateTaskInfoResponse response = client.updateTaskInfo(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateTaskInfoRequest()
        request.task_id = "{task_id}"
        request.body = MetadataCollectionTask(
        )
        response = client.update_task_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```

ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateTaskInfoRequest{}
request.TaskId = "{task_id}"
request.Body = &model.MetadataCollectionTask{
}
response, err := client.UpdateTaskInfo(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.9.5 删除单个采集任务

### 功能介绍

删除单个采集任务。

### 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/{project\_id}/metadata/tasks/{task\_id}

表 10-254 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
task_id	是	String	任务id。

## 请求参数

表 10-255 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 400

表 10-256 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-257 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**状态码： 403**

**表 10-258** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**状态码： 404**

**表 10-259** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**请求示例**

无

**响应示例**

无

**SDK 代码示例**

SDK代码示例如下。

**Java**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteTaskInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteTaskInfoRequest request = new DeleteTaskInfoRequest();
request.withTaskId("{task_id}");
try {
    DeleteTaskInfoResponse response = client.deleteTaskInfo(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteTaskInfoRequest()
        request.task_id = "{task_id}"
        response = client.delete_task_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteTaskInfoRequest{}
    request.TaskId = "{task_id}"
    response, err := client.DeleteTaskInfo(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.9.6 启动、调度、停止采集任务

### 功能介绍

启动、调度、停止采集任务。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/metadata/tasks/{task\_id}/action

表 10-260 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
task_id	是	String	任务id。

表 10-261 Query 参数

参数	是否必选	参数类型	描述
action	是	String	启动、调度、停止操作标识。

### 请求参数

表 10-262 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

### 响应参数

状态码： 400



表 10-263 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-264 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-265 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-266 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ExecuteTaskActionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ExecuteTaskActionRequest request = new ExecuteTaskActionRequest();
        request.withTaskId("{task_id}");
        try {
            ExecuteTaskActionResponse response = client.executeTaskAction(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ExecuteTaskActionRequest()
        request.task_id = "{task_id}"
        response = client.execute_task_action(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExecuteTaskActionRequest{}
    request.TaskId = "{task_id}"
    response, err := client.ExecuteTaskAction(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.9.7 获取任务日志

### 功能介绍

获取任务日志。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/metadata/tasks/{task\_id}/{instance\_id}/log

表 10-267 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
task_id	是	String	任务ID。
instance_id	是	String	任务实例ID。

## 请求参数

表 10-268 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-269 请求 Body 参数

参数	是否必选	参数类型	描述
bridge_id	是	String	桥接作业id。
classification_id	否	String	分类作业id。

## 响应参数

状态码： 200

表 10-270 响应 Body 参数

参数	参数类型	描述
enable_bridge	Boolean	是否开启桥接模式。
enable_profile	Boolean	是否启用配置。
enable_classification	Boolean	是否开启分类。
bridge_status	String	桥接状态。
profile_status	String	配置状态。
classification_status	String	分类状态。
bridge_job_log	String	桥接作业日志。
profile_job_log	String	配置作业日志。

参数	参数类型	描述
classification_job_log	String	分类作业日志。

状态码： 400

表 10-271 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-272 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-273 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-274 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "bridge_id": "D0D87362B46C4DBB8D9F7E3511E69E45",
  "classification_id": null
}
```

## 响应示例

**状态码： 200**

OK。

```
{
  "enable_bridge": true,
  "enable_profile": false,
  "enable_classification": false,
  "bridge_status": "success",
  "profile_status": null,
  "classification_status": null,
  "bridge_job_log": "Start DLI Bridge Job.",
  "profile_job_log": null,
  "classification_job_log": null
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowInstanceLogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowInstanceLogRequest request = new ShowInstanceLogRequest();
        request.withTaskId("{task_id}");
        request.withInstanceId("{instance_id}");
    }
}
```

```
JobLogRequest body = new JobLogRequest();
request.withBody(body);
try {
    ShowInstanceLogResponse response = client.showInstanceLog(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceLogRequest()
        request.task_id = "{task_id}"
        request.instance_id = "{instance_id}"
        request.body = JobLogRequest(
        )
        response = client.show_instance_log(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
)
```



```

    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceLogRequest{}
    request.TaskId = "{task_id}"
    request.InstanceId = "{instance_id}"
    request.Body = &model.JobLogRequest{
    }
    response, err := client.ShowInstanceLog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.10 数据地图接口

## 10.10.1 元数据实时同步接口(邀测)

### 功能介绍

元数据实时同步接口，支持批量。该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/metadata/async-bulk

表 10-275 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 10-276 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

表 10-277 请求 Body 参数

参数	是否必选	参数类型	描述
engine	否	String	引擎服务名称。
engine_version	否	String	引擎版本信息。
instance_id	否	String	引擎的实例ID，MRS、DWS必填。
project_id	否	String	项目ID，DLI必填。

参数	是否必选	参数类型	描述
events	否	Array of <a href="#">CatalogMetaDataEventInfo</a> objects	资产信息。

表 10-278 CatalogMetaDataEventInfo

参数	是否必选	参数类型	描述
event_ts	否	Long	事件发生时的时间戳。

参数	是否必选	参数类型	描述
event_type	否	String	事件类型， <ul style="list-style-type: none"> <li>• AddForeignKeyEvent 添加外键</li> <li>• AddNotNullConstraintEvent 添加不为空约束</li> <li>• AddPrimaryKeyEvent 添加主键</li> <li>• AddUniqueConstraintEvent 添加唯一约束</li> <li>• AlterDatabaseEvent 修改数据库</li> <li>• AddPartitionEvent 添加分区</li> <li>• AlterPartitionEvent 修改分区</li> <li>• AlterTableEvent 修改表</li> <li>• AlterCatalogEvent 修改目录</li> <li>• CreateCatalogEvent 创建目录</li> <li>• CreateDatabaseEvent 创建数据库</li> <li>• CreateFunctionEvent 创建函数</li> <li>• CreateTableEvent 创建表</li> <li>• DropConstraintEvent 删除约束</li> <li>• DropDatabaseEvent 删除数据库</li> <li>• DropFunctionEvent 删除函数</li> <li>• DropPartitionEvent 删除分区</li> <li>•.DropTableEvent 删除表</li> <li>• DropCatalogEvent 删除目录</li> <li>• AddIndexEvent 添加索引</li> <li>• AlterIndexEvent 修改索引</li> <li>• DropIndexEvent 删除索引</li> <li>• AlterSchemaEvent 修改 schema</li> <li>• CreateSchemaEvent 创建 schema</li> <li>• DropSchemaEvent 删除 schema</li> <li>• AlterColumnEvent 修改字段</li> </ul>

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> <li>• AddColumnEvent添加字段</li> <li>• DropColumnEvent删除字段</li> <li>• AlterTriggerEvent修改触发器</li> <li>• AddTriggerEvent 添加触发器</li> <li>• DropTriggerEvent删除触发器</li> </ul>
event_message	否	Object	事件消息，Map<String,Object>结构。

## 响应参数

状态码： 200

表 10-279 响应 Body 参数

参数	参数类型	描述
success	Boolean	是否成功。

状态码： 400

表 10-280 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-281 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-282 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-283 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "engine": "MRS",
  "engine_version": "3.1.2",
  "instance_id": "4bacc6b5-9e94-4938-ac3f-3c3a7cd56c01",
  "events": [ {
    "event_ts": 0,
    "event_type": "CreateTableEvent",
    "event_message": {
      "table": {
        "tableName": "testTb1"
      }
    }
  }
  ]
}
```

## 响应示例

状态码： 200

OK。

```
{
  "success": true
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class BatchSyncMetadataSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchSyncMetadataRequest request = new BatchSyncMetadataRequest();
        CatalogMetaDataSetRequest body = new CatalogMetaDataSetRequest();
        request.withBody(body);
        try {
            BatchSyncMetadataResponse response = client.batchSyncMetadata(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = BatchSyncMetadataRequest()  
    request.body = CatalogMetaDataEventRequest(  
    )  
    response = client.batch_sync_metadata(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dataartsstudio.NewDataArtsStudioClient(  
        dataartsstudio.DataArtsStudioClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.BatchSyncMetadataRequest{}  
    request.Body = &model.CatalogMetaDataEventRequest{  
    }  
    response, err := client.BatchSyncMetadata(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。



## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.10.2 搜索查询标签分页展示(邀测)

### 功能介绍

搜索查询标签分页展示，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/datamap/tags/search

表 10-284 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 10-285 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-286 请求 Body 参数

参数	是否必选	参数类型	描述
tag_ids	否	Array of strings	标签id。
type	否	String	类型。
name	否	String	标签名称，用作搜索框筛选。
create_user	否	String	创建者，用作搜索框筛选。
start	否	String	开始时间。
end	否	String	结束时间。
offset	否	Integer	页码。
limit	否	Integer	每页大小。
sort_by	否	String	根据xx排序。
sort_order	否	String	升序/降序。

## 响应参数

状态码： 200

表 10-287 响应 Body 参数

参数	参数类型	描述
limit	Integer	每页大小。
offset	Integer	页码。
count	Integer	查询出来的条数。
quota	Integer	可创建标签数量配额额。
tags	Array of <b>OpenTag</b> objects	标签实体。
total	Integer	已创建的标签总条数。

表 10-288 OpenTag

参数	参数类型	描述
name	String	标签名称。
description	String	标签描述。

参数	参数类型	描述
tag_id	String	标签id。
create_time	Long	创建时间。
update_time	Long	修改时间。
create_user	String	创建人。
domain_id	String	用户domainId。
instance_id	String	实例ID。
project_id	String	项目ID。
create_user_id	String	创建人ID。

**状态码： 400**

表 10-289 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**状态码： 401**

表 10-290 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**状态码： 403**

表 10-291 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

**状态码： 404**

表 10-292 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-293 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "offset" : 1,
  "limit" : 10,
  "sort_order" : "desc",
  "name" : "test11",
  "start" : 1685548800000,
  "end" : 1688313599999,
  "sort_by" : "createTime"
}
```

## 响应示例

状态码： 200

OK。

```
{
  "limit" : 10,
  "offset" : 1,
  "count" : 1,
  "total" : 11,
  "quota" : 100,
  "tags" : [{
    "name" : "test1123",
    "description" : "",
    "tag_id" : "8a94819c88a43c190188a4452ba40006",
    "create_time" : 1686383176612,
    "update_time" : 1686383176612,
    "create_user" : "username",
    "create_user_id" : "0833a573fe80d5401f6dc010a775569a",
    "domain_id" : "0833a5736980d53b0f22c0102ffc0",
    "instance_id" : "dd97167b873d4a79b2aad54d4370a3bc",
    "project_id" : "0833a5737480d53b2f25c010dc1a7b88"
  }]
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTagsRequest request = new ShowTagsRequest();
        TagRequest body = new TagRequest();
        request.withBody(body);
        try {
            ShowTagsResponse response = client.showTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowTagsRequest()
    request.body = TagRequest(
    )
    response = client.show_tags(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowTagsRequest{
    }
    request.Body = &model.TagRequest{
    }
    response, err := client.ShowTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.10.3 用户行为分析(邀测)

### 功能介绍

用户行为分析，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/datamap/uba

表 10-294 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-295 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID</a> 和 <a href="#">工作空间ID</a> 。

表 10-296 请求 Body 参数

参数	是否必选	参数类型	描述
event	是	String	事件名称。
params	是	Array of <a href="#">EventParam</a> objects	事件参数。

表 10-297 EventParam

参数	是否必选	参数类型	描述
guid	否	String	资产guid。
type_name	否	String	资产类型。
model_id	否	String	模块id。
property_name	否	String	参数名称。
property_value	否	String	参数值。
query	否	String	搜索框输入。
filter	否	<a href="#">DataMapFilterCriteria</a> object	查询过滤条件。
guid_list	否	Array of strings	guid列表。
trace_id	否	String	跟踪id。
source_trace_id	否	String	源跟踪id。



参数	是否必选	参数类型	描述
metadata_type_name	否	String	资产类型。
super_type_names	否	String	资产父类类型。
workspace_ids	否	Array of strings	空间列表。

表 10-298 DataMapFilterCriteria

参数	是否必选	参数类型	描述
attribute	否	String	过滤属性维度。
operator	否	String	操作标识符。
value	否	Array of strings	属性过滤值，根据attribute变化。如attribute为数据源：base.DataAsset.sourceType，则值可为["dws", "hive"]。
condition	否	String	条件拼接准则。
criterion	否	Array of <b>DataMapFilterCriteria</b> objects	条件准则。

## 响应参数

状态码： 400

表 10-299 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-300 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-301 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-302 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-303 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "event": "catalog_search_click_entity",
  "params": [{
    "query": "",
    "guid": "8050ccb6-14bc-48ea-8984-7dc661e9a2a9"
  }]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ParseUserBehaviorSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ParseUserBehaviorRequest request = new ParseUserBehaviorRequest();
        BehaviorRestBody body = new BehaviorRestBody();
        request.withBody(body);
        try {
            ParseUserBehaviorResponse response = client.parseUserBehavior(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ParseUserBehaviorRequest()
        request.body = BehaviorRestBody(
        )
        response = client.parse_user_behavior(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```

request := &model.ParseUserBehaviorRequest{
request.Body = &model.BehaviorRestBody{
}
response, err := client.ParseUserBehavior(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

### 10.10.4 资产搜索(邀测)

#### 功能介绍

资产搜索，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/datamap/entities/search

表 10-304 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-305 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <b>Token认证</b> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <b>实例ID和工作空间ID</b> 。

表 10-306 请求 Body 参数

参数	是否必选	参数类型	描述
query	否	String	查询关键字。
filter	否	<b>DataMapFilterCriteria</b> object	过滤条件对象。
facets	否	Array of strings	条件参数列表。
limit	是	Integer	分页显示每页返回结果数。默认值100。
offset	是	Integer	偏移量，默认值0。
relationship_attributes	否	Array of strings	关联关系属性。
sort	否	Array of <b>Sort</b> objects	排序信息。
owner	否	String	所有者。
query_privilege	否	Boolean	是否校验权限，默认值false。

表 10-307 DataMapFilterCriteria

参数	是否必选	参数类型	描述
attribute	否	String	过滤属性维度。
operator	否	String	操作标识符。

参数	是否必选	参数类型	描述
value	否	Array of strings	属性过滤值，根据attribute变化。如attribute为数据源：base.DataAsset.sourceType，则值可为["dws", "hive"]。
condition	否	String	条件拼接准则。
criterion	否	Array of <b>DataMapFilterCriteria</b> objects	条件准则。

表 10-308 Sort

参数	是否必选	参数类型	描述
attribute	否	String	属性。
order	否	String	排序枚举值，默认值ASC。

## 响应参数

状态码： 200

表 10-309 响应 Body 参数

参数	参数类型	描述
count	Number	查询结果总量。
entities	Array of <b>Entity</b> objects	资产实体列表。
facets	Array of objects	资产分类facets维度信息列表，数据结构 List<Map<String, List<Aggregation>>> 取值为 count。
metrics	Object	资产分类metrics维度信息列表，数据结构 List<Map<String, List<Aggregation>>> 取值为 aggregation。
referred_entities	Object	关联资产，数据类型Map<String, Entity>， key: guid, value: 资产信息。

表 10-310 Entity

参数	参数类型	描述
version	Number	数据版本。
relationship_attributes	Object	关联关系属性，数据类型Map<String, Object>，key: 关系类型: value: 关联关系对象。
super_type_names	Array of strings	父类资产类型。
business_attributes	Object	业务属性，数据类型Map<String, Map<String, Object>>，key: 业务关系类型, value: 业务关系。
multi_attributes	Object	承担密级和标签的多值对象数据结构，数据结构Map<String, List<Map<String, Object>>>，key: 综合关系类型, value: 综合关系。
privilege_info	<b>EntityPrivilegeInfo</b> object	实体特权信息。
extended_attributes	Object	拓展属性，数据结构Map<String, Object>，key: 拓展属性, value: 拓展属性值。
guid	String	资产guid。
type_name	String	资产类型名称。
type_display_name	String	类型展示名称。
display_text	String	展示名称。
attributes	Object	资产属性，Map<String, Object>。key: 属性, value: 属性值。
updated_attributes	Array of strings	修改属性列表。
project_id	String	项目id。
domain_id	String	主账号id。
instance_id	Array of strings	实例化id。
workspace_id	Array of strings	空间id列表。
status	String	状态。
created_by	String	创建人。
updated_by	String	修改人。
create_time	String	创建时间。
update_time	String	修改时间。



表 10-311 EntityPrivilegeInfo

参数	参数类型	描述
privileges	Array of strings	特权列表。
inherit_privileges	Array of strings	继承特权列表。

状态码： 400

表 10-312 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-313 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-314 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-315 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-316 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "query": "",
  "filter": {
    "attribute": null,
    "condition": "AND",
    "criterion": [ {
      "attribute": null,
      "condition": "OR",
      "criterion": [ {
        "attribute": "superTypeNames",
        "operator": "IN",
        "value": [ "dataasset.Database" ]
      }, {
        "attribute": "typeName",
        "operator": "IN",
        "value": [ "dataasset.Database" ]
      } ]
    }, {
      "attribute": "base.DataAsset.sourceType",
      "operator": "IN",
      "value": [ "dli" ]
    } ]
  },
  "facets": [ "category.path", "typeName", "base.DataAsset.sourceType", "classifications.name",
    "tags.name", "securityLevel.name", "workspaceId", "base.Asset.catalogId" ],
  "limit": 10,
  "offset": 1,
  "sort": [ {
    "attribute": "createTime",
    "order": "DESC"
  } ]
}
```

## 响应示例

状态码： 200

OK。

```
{
  "count": 1234,
```

```

"entities" : [ {
  "guid" : "a14d047e-3d67-4f71-90f0-53b6759a7fa5",
  "attributes" : {
    "owner" : "username",
    "is_favorite" : 0,
    "qualifiedName" : "dws@a96be22c-77f3-46ea-96ab-c8913052ec7d",
    "favorites_count" : 0,
    "description" : "This temp entity was generated from DLF lineage service",
    "clusterId" : "",
    "click_through" : 10,
    "baseType" : "Source",
    "path" : "",
    "catalogId" : "",
    "workspaceVslid" : { },
    "createTime" : "2022-12-09T01:59:42.319+00:00",
    "port" : 8000,
    "sourceType" : "dws",
    "name" : "dws_4autotest1_nomodify",
    "host" : "",
    "id" : "f9338475eb8c4cbda61b85c05908a432",
    "projectId" : ""
  },
  "status" : "ACTIVE",
  "version" : 0,
  "type_name" : "dataasset.Source",
  "type_display_name" : "数据源",
  "project_id" : 123456,
  "domain_id" : 123456,
  "instance_id" : [ "123456" ],
  "workspace_id" : [ "123456" ],
  "updated_by" : "username",
  "create_time" : "2023-05-17T06:32:00.511+00:00",
  "update_time" : "2023-05-17T06:32:00.511+00:00",
  "super_type_names" : [ "base.Asset", "base.Infrastructure" ],
  "business_attributes" : {
    "businessName" : {
      "confidence" : 1.0,
      "name" : "dws0505",
      "confirmed" : true
    }
  },
  "multi_attributes" : {
    "tags" : [ {
      "confidence" : 1.0,
      "name" : "个人数据误删",
      "confirmed" : false
    } ]
  },
  "privilege_info" : {
    "privileges" : [ ],
    "inherit_privileges" : [ ]
  },
  "extended_attributes" : { },
  "display_text" : "dws0505"
} ],
"facets" : [ {
  "typeName" : [ {
    "value" : "dataasset.Table",
    "id" : "dataasset.Table",
    "count" : 816,
    "aggregation" : 0.0,
    "children" : [ ]
  }, {
    "value" : "dlm.API",
    "id" : "dlm.API",
    "count" : 103,
    "aggregation" : 0.0,
    "children" : [ ]
  } ]
} ]

```

```

}, {
  "base.DataAsset.sourceType" : [ {
    "value" : "dli",
    "id" : "dli",
    "count" : 1,
    "aggregation" : 0.0,
    "children" : [ ]
  }, {
    "value" : "hive",
    "id" : "hive",
    "count" : 1,
    "aggregation" : 0.0,
    "children" : [ ]
  } ]
}, {
  "classifications.name" : [ {
    "value" : "01d1d582b14c4dfe8ebe53b15d88ffc2/c1549ae148144a37b219344892a5568f",
    "id" : "01d1d582b14c4dfe8ebe53b15d88ffc2/c1549ae148144a37b219344892a5568f",
    "count" : 2,
    "aggregation" : 0.0,
    "children" : [ ]
  }, {
    "value" : "01d1d582b14c4dfe8ebe53b15d88ffc2/e1893cd5baf34182b9d2e2728a02a2a8",
    "id" : "01d1d582b14c4dfe8ebe53b15d88ffc2/e1893cd5baf34182b9d2e2728a02a2a8",
    "count" : 2,
    "aggregation" : 0.0,
    "children" : [ ]
  } ]
}, {
  "tags.name" : [ {
    "value" : "tag_testmLEi",
    "id" : "tag_testmLEi",
    "count" : 2,
    "aggregation" : 0.0,
    "children" : [ ]
  }, {
    "value" : "个人数据误删",
    "id" : "个人数据误删",
    "count" : 2,
    "aggregation" : 0.0,
    "children" : [ ]
  } ]
}, {
  "securityLevel.name" : [ {
    "value" : "8a9480037c747038017c77363e510000",
    "id" : "8a9480037c747038017c77363e510000",
    "count" : 2,
    "aggregation" : 0.0,
    "children" : [ ]
  } ]
}, {
  "workspaceId" : [ {
    "value" : "1b59d3c777ad4d619b89eeac4f3cce87",
    "id" : "1b59d3c777ad4d619b89eeac4f3cce87",
    "count" : 1234,
    "aggregation" : 0.0,
    "children" : [ ]
  } ]
}, {
  "category.path" : [ {
    "value" : "ceshi_test",
    "id" : "eeb055e69c624311b6b9cfee89a4ec70/ceshi_test",
    "count" : 1234,
    "aggregation" : 0.0,
    "children" : [ ]
  }, {
    "value" : "交通",
    "id" : "1b59d3c777ad4d619b89eeac4f3cce87/交通",
    "count" : 1234,
  } ]
} ]

```

```
"aggregation": 0.0,
"children": [ {
  "value": "行程记录",
  "id": "1b59d3c777ad4d619b89eeac4f3cce87/交通/行程记录",
  "count": 1234,
  "aggregation": 0.0,
  "children": [ {
    "value": "行程事实",
    "id": "1b59d3c777ad4d619b89eeac4f3cce87/交通/行程记录/行程事实",
    "count": 1234,
    "aggregation": 0.0,
    "children": [ ]
  } ]
} ]
} ]
} ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowDataSetsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDataSetsRequest request = new ShowDataSetsRequest();
        SearchParameter body = new SearchParameter();
        request.withBody(body);
        try {
            ShowDataSetsResponse response = client.showDataSets(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowDataSetsRequest()
        request.body = SearchParameter(
        )
        response = client.show_data_sets(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowDataSetsRequest{}
request.Body = &model.SearchParameter{
}
response, err := client.ShowDataSets(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.10.5 创建或修改资产(邀测)

### 功能介绍

创建或修改资产，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/datamap/entities

表 10-317 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 10-318 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-319 请求 Body 参数

参数	是否必选	参数类型	描述
entity	否	<a href="#">Entity</a> object	资产信息。
referred_entities	否	Object	关联资产信息，结构体 Map<String, Entity>。key: 资产guid, value: 资产信息。

表 10-320 Entity

参数	是否必选	参数类型	描述
version	否	Number	数据版本。
relationship_attributes	否	Object	关联关系属性，数据类型 Map<String, Object>, key: 关系类型: value: 关联关系对象。
super_type_names	否	Array of strings	父类资产类型。
business_attributes	否	Object	业务属性，数据类型 Map<String, Map<String, Object>>, key: 业务关系类型, value: 业务关系。



参数	是否必选	参数类型	描述
multi_attributes	否	Object	承担密级和标签的多值对象数据结构，数据结构Map<String, List<Map<String, Object>>>，key: 综合关系类型，value: 综合关系。
privilege_info	否	<b>EntityPrivilegeInfo</b> object	实体特权信息。
extended_attributes	否	Object	拓展属性，数据结构Map<String, Object>，key: 拓展属性，value: 拓展属性值。
guid	否	String	资产guid。
type_name	否	String	资产类型名称。
type_display_name	否	String	类型展示名称。
display_text	否	String	展示名称。
attributes	否	Object	资产属性，Map<String, Object>。key: 属性，value: 属性值。
updated_attributes	否	Array of strings	修改属性列表。
project_id	否	String	项目id。
domain_id	否	String	主账号id。
instance_id	否	Array of strings	实例化id。
workspace_id	否	Array of strings	空间id列表。
status	否	String	状态。
created_by	否	String	创建人。
updated_by	否	String	修改人。
create_time	否	String	创建时间。
update_time	否	String	修改时间。

表 10-321 EntityPrivilegeInfo

参数	是否必选	参数类型	描述
privileges	否	Array of strings	特权列表。
inherit_privileges	否	Array of strings	继承特权列表。

## 响应参数

状态码： 400

表 10-322 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-323 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-324 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-325 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-326 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "entity": {
    "guid": -1,
    "attributes": {
      "owner": "username",
      "is_favorite": 0,
      "qualifiedName": "dws@a96be22c-77f3-46ea-96ab-c8913052ec7d",
      "favorites_count": 0,
      "description": "This temp entity was generated from DLF lineage service",
      "clusterId": "",
      "click_through": 10,
      "baseType": "Source",
      "path": "",
      "catalogId": "",
      "workspaceVslId": { },
      "createTime": 0,
      "port": 8000,
      "sourceType": "dws",
      "name": "dws_4autotest1_nomodify",
      "host": "",
      "id": "f9338475eb8c4cbda61b85c05908a432",
      "projectId": ""
    },
    "status": "ACTIVE",
    "version": 0,
    "type_name": "dataasset.Source",
    "type_display_name": "数据源",
    "project_id": 123456,
    "domain_id": 123456,
    "instance_id": [ "123456" ],
    "workspace_id": [ "123456" ],
    "updated_by": "username",
    "create_time": 0,
    "update_time": 0,
    "super_type_names": [ "base.Asset", "base.Infrastructure" ],
    "business_attributes": {
      "businessName": {
        "confidence": 1.0,
        "name": "dws0505",
        "confirmed": true
      }
    }
  }
}
```

```
    },
    "multi_attributes" : {
      "tags" : [ {
        "confidence" : 1.0,
        "name" : "个人数据误删",
        "confirmed" : false
      } ]
    },
    "privilege_info" : {
      "privileges" : [ ],
      "inherit_privileges" : [ ]
    },
    "extended_attributes" : { },
    "display_text" : "dws0505"
  },
  "referred_entities" : null
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateOrUpdateEntitiesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateOrUpdateEntitiesRequest request = new CreateOrUpdateEntitiesRequest();
        DataEntityWithExtInfo body = new DataEntityWithExtInfo();
        request.withBody(body);
        try {
            CreateOrUpdateEntitiesResponse response = client.createOrUpdateEntities(request);
            System.out.println(response.toString());
        }
    }
}
```

```
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateOrUpdateEntitiesRequest()
        request.body = DataEntityWithExtInfo(
        )
        response = client.create_or_update_entities(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateOrUpdateEntitiesRequest{}
request.Body = &model.DataEntityWithExtInfo{
}
response, err := client.CreateOrUpdateEntities(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.10.6 资产详情(邀测)

### 功能介绍

资产详情接口，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/datamap/entities/guid/{guid}

表 10-327 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

表 10-328 Query 参数

参数	是否必选	参数类型	描述
ignore_relationships	否	Boolean	是否忽略关联资产，取值范围：true（忽略关联资产）、false（不忽略关联资产）缺省值：false。

## 请求参数

表 10-329 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-330 响应 Body 参数

参数	参数类型	描述
entity	<a href="#">Entity</a> object	资产信息。

参数	参数类型	描述
referred_entities	Object	关联资产信息，结构体Map<String, Entity>。 key: 资产guid, value: 资产信息。

表 10-331 Entity

参数	参数类型	描述
version	Number	数据版本。
relationship_attributes	Object	关联关系属性，数据类型Map<String, Object>， key: 关系类型: value: 关联关系对象。
super_type_names	Array of strings	父类资产类型。
business_attributes	Object	业务属性，数据类型Map<String, Map<String, Object>>， key: 业务关系类型, value: 业务关系。
multi_attributes	Object	承担密级和标签的多值对象数据结构，数据结构 Map<String, List<Map<String, Object>>>， key: 综合关系类型, value: 综合关系。
privilege_info	<b>EntityPrivilegeInfo</b> object	实体特权信息。
extended_attributes	Object	拓展属性，数据结构Map<String, Object>， key: 拓展属性, value: 拓展属性值。
guid	String	资产guid。
type_name	String	资产类型名称。
type_display_name	String	类型展示名称。
display_text	String	展示名称。
attributes	Object	资产属性，Map<String, Object>。key: 属性， value: 属性值。
updated_attributes	Array of strings	修改属性列表。
project_id	String	项目id。
domain_id	String	主账号id。
instance_id	Array of strings	实例化id。
workspace_id	Array of strings	空间id列表。



参数	参数类型	描述
status	String	状态。
created_by	String	创建人。
updated_by	String	修改人。
create_time	String	创建时间。
update_time	String	修改时间。

表 10-332 EntityPrivilegeInfo

参数	参数类型	描述
privileges	Array of strings	特权列表。
inherit_privileges	Array of strings	继承特权列表。

状态码： 400

表 10-333 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-334 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-335 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-336 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-337 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
{
  "entity": {
    "guid": "a14d047e-3d67-4f71-90f0-53b6759a7fa5",
    "attributes": {
      "owner": "username",
      "is_favorite": 0,
      "qualifiedName": "dws@a96be22c-77f3-46ea-96ab-c8913052ec7d",
      "favorites_count": 0,
      "description": "This temp entity was generated from DLF lineage service",
      "clusterId": "",
      "click_through": 10,
      "baseType": "Source",
      "path": "",
      "catalogId": "",
      "workspaceVslId": { },
      "createTime": "2022-12-09T01:59:42.319+00:00",

```

```
"port" : 8000,
"sourceType" : "dws",
"name" : "dws_4autotest1_nomodify",
"host" : "",
"id" : "f9338475eb8c4cbda61b85c05908a432",
"projectId" : ""
},
"status" : "ACTIVE",
"version" : 0,
"type_name" : "dataasset.Source",
"type_display_name" : "数据源",
"project_id" : 123456,
"domain_id" : 123456,
"instance_id" : [ "123456" ],
"workspace_id" : [ "123456" ],
"updated_by" : "username",
"create_time" : "2023-05-17T06:32:00.511+00:00",
"update_time" : "2023-05-17T06:32:00.511+00:00",
"super_type_names" : [ "base.Asset", "base.Infrastructure" ],
"business_attributes" : {
  "businessName" : {
    "confidence" : 1.0,
    "name" : "dws0505",
    "confirmed" : true
  }
},
"multi_attributes" : {
  "tags" : [ {
    "confidence" : 1.0,
    "name" : "个人数据误删",
    "confirmed" : false
  } ]
},
"privilege_info" : {
  "privileges" : [ ],
  "inherit_privileges" : [ ]
},
"extended_attributes" : { },
"display_text" : "dws0505"
},
"referred_entities" : null
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowDataDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
```

this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowDataDetailRequest request = new ShowDataDetailRequest();
request.withGuid("{guid}");
try {
    ShowDataDetailResponse response = client.showDataDetail(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowDataDetailRequest()
        request.guid = "{guid}"
        response = client.show_data_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDataDetailRequest{}
    request.Guid = "{guid}"
    response, err := client.ShowDataDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。

状态码	描述
404	Not Found。
500	InternalServerError。

## 10.10.7 资产血缘(邀测)

### 功能介绍

资产血缘接口，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/datamap/lineage/guid/{guid}

表 10-338 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

表 10-339 Query 参数

参数	是否必选	参数类型	描述
direction	否	String	血缘查询方向，默认值：BOTH。
relationship_types	否	Array of strings	关联关系类型列表，血缘查询使用默认值空
relationship_type_categories	否	Array of strings	关联关系类型类别，默认空。血缘查询使用DATA_FLOW
related_entity_types	否	Array of strings	关联实体类型，指定上下游资产类型，血缘查询使用默认值空。
extend_processes_data_flow	否	Boolean	是否扩展数据，血缘查询使用默认值false。

参数	是否必选	参数类型	描述
include_parent_entity	否	Boolean	是否包含父类实体，默认 false。

## 请求参数

表 10-340 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID</a> 和 <a href="#">工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-341 响应 Body 参数

参数	参数类型	描述
total	Number	查询命中总条数。
relationships	Array of <a href="#">Lineage</a> objects	关系列表。
relationship_types	Object	关系类型。
entities	Array of <a href="#">Entity</a> objects	资产信息。
entity_types	Object	实体类型。

表 10-342 Lineage

参数	参数类型	描述
relation_type_name	String	关系类型。 PARENT_CHILD, LOGICAL_PHYSICAL, PK_FK, DATA_FLOW, INSTANCE_OF, JOIN, IS_A, UP_DOWN, ASSOCIATION, WORK_FLOW。
direction	String	血缘流向。
ep1_id	String	节点一资产guid。
ep1_type_name	String	节点一资产类型。
ep2_id	String	节点二资产guid。
ep2_type_name	String	节点二资产类型。
end1	<b>Entity</b> object	节点一资产。
end2	<b>Entity</b> object	节点二资产。
propagate_tag	String	关系类型。
guid	String	资产guid。
type_name	String	资产类型名称。
type_display_name	String	类型展示名称。
display_text	String	展示名称。
attributes	Object	资产属性。Map结构, key: 属性名称, value: 属性值
updated_attributes	Array of strings	修改属性列表。
project_id	String	项目id。
domain_id	String	主账号id。
instance_id	Array of strings	实例id。
workspace_id	Array of strings	空间id列表。
status	String	状态。
created_by	String	创建人。
updated_by	String	修改人。
create_time	String	创建时间。



参数	参数类型	描述
update_time	String	修改时间。

表 10-343 Entity

参数	参数类型	描述
version	Number	数据版本。
relationship_attributes	Object	关联关系属性，数据类型Map<String, Object>，key: 关系类型: value: 关联关系对象。
super_type_names	Array of strings	父类资产类型。
business_attributes	Object	业务属性，数据类型Map<String, Map<String, Object>>，key: 业务关系类型，value: 业务关系。
multi_attributes	Object	承担密级和标签的多值对象数据结构，数据结构Map<String, List<Map<String, Object>>>，key: 综合关系类型，value: 综合关系。
privilege_info	<b>EntityPrivilegeInfo</b> object	实体特权信息。
extended_attributes	Object	拓展属性，数据结构Map<String, Object>，key: 拓展属性，value: 拓展属性值。
guid	String	资产guid。
type_name	String	资产类型名称。
type_display_name	String	类型展示名称。
display_text	String	展示名称。
attributes	Object	资产属性，Map<String, Object>。key: 属性，value: 属性值。
updated_attributes	Array of strings	修改属性列表。
project_id	String	项目id。
domain_id	String	主账号id。
instance_id	Array of strings	实例化id。
workspace_id	Array of strings	空间id列表。
status	String	状态。

参数	参数类型	描述
created_by	String	创建人。
updated_by	String	修改人。
create_time	String	创建时间。
update_time	String	修改时间。

表 10-344 EntityPrivilegeInfo

参数	参数类型	描述
privileges	Array of strings	特权列表。
inherit_privileges	Array of strings	继承特权列表。

状态码： 400

表 10-345 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-346 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-347 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

状态码： 404

表 10-348 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-349 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
{
  "total": 1,
  "relationships": [ {
    "guid": "62c8a705-abe8-4d25-b974-854a6b97343c",
    "status": "ACTIVE",
    "direction": "IN",
    "type_name": "dataasset.tables_schema",
    "type_display_name": "数据表:模式",
    "project_id": "0833a5737480d53b2f25c010dc1a7b88",
    "instance_id": [ "dd97167b873d4a79b2aad54d4370a3bc" ],
    "workspace_id": [ "ae559f076ac14061921b6c6c331576c4", "8f91082a19474825b2fee2147e9b464c",
    "1b59d3c777ad4d619b89eeac4f3cce87" ],
    "created_by": "username",
    "updated_by": "username",
    "create_time": "2024-01-03T09:11:42.871+00:00",
    "update_time": "2024-01-04T11:52:36.454+00:00",
    "relation_type_name": "PARENT_CHILD",
    "ep1_id": "0c0f000e-d7b2-4aab-956c-cdc8b07083c6",
    "ep2_id": "6a1510a9-f40c-477f-ae3a-638e592b6504",
    "ep1_type_name": "dws.Table",
```

```

"ep2_type_name" : "dws.Schema",
"propagate_tag" : "NONE"
}, {
  "guid" : "64d40028-10b9-41ed-811e-9c32689983eb",
  "status" : "ACTIVE",
  "direction" : "IN",
  "type_name" : "dataasset.tables_database",
  "type_display_name" : "数据表:数据库",
  "project_id" : "0833a5737480d53b2f25c010dc1a7b88",
  "instance_id" : [ "dd97167b873d4a79b2aad54d4370a3bc" ],
  "workspace_id" : [ "ae559f076ac14061921b6c6c331576c4", "8f91082a19474825b2fee2147e9b464c",
"1b59d3c777ad4d619b89eeac4f3cce87" ],
  "created_by" : "username",
  "updated_by" : "username",
  "create_time" : "2024-01-03T09:11:42.872+00:00",
  "update_time" : "2024-01-04T11:52:36.454+00:00",
  "relation_type_name" : "PARENT_CHILD",
  "ep1_id" : "0c0f000e-d7b2-4aab-956c-cdc8b07083c6",
  "ep2_id" : "e10623fb-dbfc-4f43-bccc-33c68fb4c774",
  "ep1_type_name" : "dws.Table",
  "ep2_type_name" : "dws.Database",
  "propagate_tag" : "NONE"
}, {
  "guid" : "cfdd16e6-3827-4572-8b16-d200bb5fcc79",
  "attributes" : {
    "__index" : 1,
    "__key" : null
  },
  "status" : "ACTIVE",
  "direction" : "OUT",
  "type_name" : "process.node_job",
  "type_display_name" : "作业:节点",
  "project_id" : "0833a5737480d53b2f25c010dc1a7b88",
  "instance_id" : [ "dd97167b873d4a79b2aad54d4370a3bc" ],
  "workspace_id" : [ "1b59d3c777ad4d619b89eeac4f3cce87" ],
  "created_by" : "username",
  "updated_by" : "username",
  "create_time" : "2024-01-03T10:17:24.696+00:00",
  "update_time" : "2024-01-03T10:17:24.696+00:00",
  "relation_type_name" : "PARENT_CHILD",
  "ep1_id" : "71e0b8d8-abc8-4cf0-94ba-e8e0e04c9b5a",
  "ep2_id" : "806e2242-a579-4340-aed0-7c8f35ec52fa",
  "ep1_type_name" : "dlf.Job",
  "ep2_type_name" : "dlf.dws_sql",
  "propagate_tag" : "NONE"
}, {
  "guid" : "8ab58411-db50-477f-b0e8-3faacefb3216",
  "attributes" : {
    "__index" : 0,
    "__key" : null
  },
  "status" : "ACTIVE",
  "direction" : "OUT",
  "type_name" : "base.process_dataset_outputs",
  "type_display_name" : "输出",
  "project_id" : "0833a5737480d53b2f25c010dc1a7b88",
  "instance_id" : [ "dd97167b873d4a79b2aad54d4370a3bc" ],
  "workspace_id" : [ "1b59d3c777ad4d619b89eeac4f3cce87" ],
  "created_by" : "username",
  "updated_by" : "username",
  "create_time" : "2024-01-03T10:17:34.823+00:00",
  "update_time" : "2024-01-03T10:17:34.823+00:00",
  "relation_type_name" : "DATA_FLOW",
  "ep1_id" : "806e2242-a579-4340-aed0-7c8f35ec52fa",
  "ep2_id" : "0c0f000e-d7b2-4aab-956c-cdc8b07083c6",
  "ep1_type_name" : "dlf.dws_sql",
  "ep2_type_name" : "dws.Table",
  "propagate_tag" : "NONE"
}
],

```

```

"relationship_types": null,
"entities": [ {
  "guid": "71e0b8d8-abc8-4cf0-94ba-e8e0e04c9b5a",
  "attributes": {
    "owner": "username",
    "path": "",
    "catalogId": "catalog@cn-north-7",
    "scheduleType": "REAL_TIME",
    "createTime": "2024-01-03T10:17:17.924+00:00",
    "qualifiedName": "19424819@dlf_job.0833a5737480d53b2f25c010dc1a7b88-
workspace-1b59d3c777ad4d619b89eeac4f3cce87",
    "name": "job_6757",
    "description": "BATCH",
    "id": 19424819,
    "priority": 0,
    "cronExp": "",
    "baseType": "Job"
  },
  "status": "ACTIVE",
  "version": 5,
  "type_name": "dlf.Job",
  "type_display_name": "作业",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "domain_id": "0833a5736980d53b0f22c0102ffc0",
  "instance_id": [ "dd97167b873d4a79b2aad54d4370a3bc" ],
  "workspace_id": [ "1b59d3c777ad4d619b89eeac4f3cce87" ],
  "created_by": "username",
  "updated_by": "username",
  "create_time": "2024-01-03T10:12:07.391+00:00",
  "update_time": "2024-01-03T10:17:24.696+00:00",
  "super_type_names": [ "base.Process", "base.Asset", "process.Job" ],
  "business_attributes": {
    "businessName": {
      "confidence": 1,
      "name": "job_6757",
      "confirmed": true
    }
  },
  "multi_attributes": {
    "tags": [ ]
  },
  "extended_attributes": { },
  "display_text": "job_6757"
}, {
  "guid": "0c0f000e-d7b2-4aab-956c-cdc8b07083c6",
  "attributes": {
    "sourceId": "dws@94a1ad37-1bde-4919-8e93-712577a9c3b2",
    "description": "",
    "ddlCreateTime": 1691478978133,
    "dataConnectionId": [ "3653cce08ed7433bad507a30d86f2f2c" ],
    "totalFrequency": 0,
    "baseType": "Table",
    "tableType": "BASE TABLE",
    "partitionCount": 0,
    "path": "/wk/public",
    "hasRules": false,
    "catalogId": "catalog@cn-north-7",
    "writeFrequency": 0,
    "schemaQName": "wk.public@dws.94a1ad37-1bde-4919-8e93-712577a9c3b2",
    "isPartition": false,
    "tableSize": 0,
    "releaseStatus": "",
    "alias": "",
    "hasTriggers": false,
    "dataUpdateTime": 0,
    "publishStatus": "",
    "owner": "dbadmin",
    "orientation": "ROW",
    "lastAccessTime": 0,

```

```

"qualifiedName" : "wk.public.wk_02@dws.94a1ad37-1bde-4919-8e93-712577a9c3b2",
"qualityScoreOverall" : 0,
"readFrequency" : 0,
"partitionType" : "",
"sourceType" : "dws",
"rowCounts" : 0,
"name" : "wk_02",
"databaseQName" : "wk@dws.94a1ad37-1bde-4919-8e93-712577a9c3b2",
"location" : "",
"comment" : "",
"partitionKeyCount" : 0,
"sourceName" : "dws_nossl_4autotest_nomodify",
"hasIndexes" : false,
"ddlUpdateTime" : 1691478978133
},
"status" : "ACTIVE",
"version" : 2,
"type_name" : "dws.Table",
"type_display_name" : "数据表",
"project_id" : "0833a5737480d53b2f25c010dc1a7b88",
"domain_id" : "0833a5736980d53b0f22c0102ffcbfc0",
"instance_id" : [ "dd97167b873d4a79b2aad54d4370a3bc" ],
"workspace_id" : [ "ae559f076ac14061921b6c6c331576c4", "1b59d3c777ad4d619b89eeac4f3cce87" ],
"created_by" : "username",
"updated_by" : "username",
"create_time" : "2024-01-03T09:11:42.871+00:00",
"update_time" : "2024-01-04T11:52:36.454+00:00",
"super_type_names" : [ "base.DataAsset", "dataasset.Table", "base.DataSet", "base.Asset" ],
"business_attributes" : {
  "businessName" : {
    "confidence" : 1,
    "name" : "wk_02",
    "confirmed" : true
  }
},
"multi_attributes" : {
  "classifications" : [ ],
  "tags" : [ ]
},
"extended_attributes" : { },
"display_text" : "wk_02"
}, {
  "guid" : "6a1510a9-f40c-477f-ae3a-638e592b6504",
  "attributes" : {
    "sourceId" : "dws@94a1ad37-1bde-4919-8e93-712577a9c3b2",
    "owner" : "Ruby",
    "path" : "/wk",
    "catalogId" : "catalog@cn-north-7",
    "sourceType" : "dws",
    "qualifiedName" : "wk.public@dws.94a1ad37-1bde-4919-8e93-712577a9c3b2",
    "name" : "public",
    "databaseQName" : "wk@dws.94a1ad37-1bde-4919-8e93-712577a9c3b2",
    "description" : "gs_roach_stop_backup",
    "sourceName" : "dws_nossl_4autotest_nomodify",
    "dataConnectionId" : [ "3653cce08ed7433bad507a30d86f2f2c" ],
    "baseType" : "Schema"
  },
  "status" : "ACTIVE",
  "version" : 3,
  "type_name" : "dws.Schema",
  "type_display_name" : "模式",
  "project_id" : "0833a5737480d53b2f25c010dc1a7b88",
  "domain_id" : "0833a5736980d53b0f22c0102ffcbfc0",
  "instance_id" : [ "dd97167b873d4a79b2aad54d4370a3bc" ],
  "workspace_id" : [ "8f91082a19474825b2fee2147e9b464c", "ae559f076ac14061921b6c6c331576c4", "1b59d3c777ad4d619b89eeac4f3cce87" ],
  "created_by" : "username",
  "updated_by" : "username",
  "create_time" : "2024-01-02T08:43:44.775+00:00",

```

```

"update_time" : "2024-01-04T11:52:26.054+00:00",
"super_type_names" : [ "base.DataAsset", "dataasset.Schema", "base.Asset" ],
"business_attributes" : {
  "businessName" : {
    "confidence" : 1,
    "name" : "public",
    "confirmed" : true
  }
},
"multi_attributes" : {
  "tags" : [ ]
},
"extended_attributes" : { },
"display_text" : "public"
}, {
"guid" : "e10623fb-dbfc-4f43-bccc-33c68fb4c774",
"attributes" : {
  "sourceId" : "dws@94a1ad37-1bde-4919-8e93-712577a9c3b2",
  "owner" : "dbadmin",
  "qualifiedName" : "wk@dws.94a1ad37-1bde-4919-8e93-712577a9c3b2",
  "description" : "",
  "encoding" : "SQL_ASCII",
  "dataConnectionId" : [ "3653cce08ed7433bad507a30d86f2f2c" ],
  "baseType" : "Database",
  "path" : "/",
  "catalogId" : "catalog@cn-north-7",
  "sourceType" : "dws",
  "isTemplate" : false,
  "name" : "wk",
  "comment" : "",
  "sourceName" : "dws_nossl_4autotest_nomodify"
},
"status" : "ACTIVE",
"version" : 3,
"type_name" : "dws.Database",
"type_display_name" : "数据库",
"project_id" : "0833a5737480d53b2f25c010dc1a7b88",
"domain_id" : "0833a5736980d53b0f22c0102ffcbfc0",
"instance_id" : [ "dd97167b873d4a79b2aad54d4370a3bc" ],
"workspace_id" : [ "8f91082a19474825b2fee2147e9b464c", "ae559f076ac14061921b6c6c331576c4",
"1b59d3c777ad4d619b89eeac4f3cce87" ],
"created_by" : "username",
"updated_by" : "username",
"create_time" : "2024-01-02T08:43:44.726+00:00",
"update_time" : "2024-01-04T11:52:25.995+00:00",
"super_type_names" : [ "base.DataAsset", "base.Asset", "dataasset.Database" ],
"business_attributes" : {
  "businessName" : {
    "confidence" : 1,
    "name" : "wk",
    "confirmed" : true
  }
},
"multi_attributes" : {
  "tags" : [ ]
},
"extended_attributes" : { },
"display_text" : "wk"
}, {
"guid" : "806e2242-a579-4340-aed0-7c8f35ec52fa",
"attributes" : {
  "owner" : "username",
  "scriptPath" : "8abfdb5a8cce4a43018cced3039200c2",
  "statementOrScript" : "SCRIPT",
  "qualifiedName" : "19424819.e2b86e34-3223-4332-ac24-
c61f9af72e22@dlf_job.0833a5737480d53b2f25c010dc1a7b88-workspace
-1b59d3c777ad4d619b89eeac4f3cce87",
  "failRetry" : false,
  "description" : "BATCH",

```

```

"pollingTime" : 20,
"nodeType" : "dws_sql_node",
"timeout" : 360,
"baseType" : "Node",
"args" : null,
"failPolicy" : "EXIT",
"path" : null,
"database" : "wk",
"catalogId" : "catalog@cn-north-7",
"maxRetryTime" : null,
"name" : "xxxx",
"statement" : "-- DWS sql -- ***** ---- author:
username-- create time: 2024/01/03 18:16:15 GMT+08:00--
***** --INSERT INTO wk_02 SELECT * FROM wk_01;",
"retryInterval" : null,
"connection" : "a0b248d34ab84631872f46d46a44f0b2",
"dirtyDataPath" : null
},
"status" : "ACTIVE",
"version" : 1,
"type_name" : "dlf.dws_sql",
"type_display_name" : "节点",
"project_id" : "0833a5737480d53b2f25c010dc1a7b88",
"domain_id" : "0833a5736980d53b0f22c0102ffcbfc0",
"instance_id" : [ "dd97167b873d4a79b2aad54d4370a3bc" ],
"workspace_id" : [ "1b59d3c777ad4d619b89eeac4f3cce87" ],
"created_by" : "username",
"updated_by" : "username",
"create_time" : "2024-01-03T10:17:24.696+00:00",
"update_time" : "2024-01-03T10:17:24.696+00:00",
"super_type_names" : [ "base.Process", "dlf.Node", "base.Asset", "process.Node" ],
"business_attributes" : {
  "businessName" : {
    "confidence" : 1,
    "name" : "xxxx",
    "confirmed" : true
  }
},
"multi_attributes" : {
  "tags" : [ ]
},
"extended_attributes" : { },
"display_text" : "xxxx"
}],
"entity_types" : null
}

```

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。



## 10.10.8 批量血缘(邀测)

### 功能介绍

批量血缘接口，一次性获取所有作业算子的血缘。该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/datamap/lineage/bulk

表 10-350 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 10-351 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页参数偏移量，默认值0。
limit	否	Integer	分页参每页数量，默认值100。

### 请求参数

表 10-352 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

### 响应参数

状态码： 200

表 10-353 响应 Body 参数

参数	参数类型	描述
total	Number	作业算子个数，批量查询根据作业算子获取血缘。
data	Array of <a href="#">TableLineageInfo</a> objects	当前页作业算子包含的表血缘列表

表 10-354 TableLineageInfo

参数	参数类型	描述
source_guid	String	上游血缘资产guid。
source_qualified_name	String	上游血缘资产Qname。
source_type	String	上游血缘资产类型。
source_db	String	上游血缘资产数据库。
source_schema	String	上游血缘资产逻辑库。
source_table	String	上游血缘资产表。
target_guid	String	下游血缘资产guid。
target_qualified_name	String	下游血缘资产Qname。
target_type	String	下游血缘资产类型。
target_db	String	下游血缘资产数据库。
target_schema	String	下游血缘资产逻辑库。
target_table	String	下游血缘资产表。
node_guid	String	作业算子guid。
node_name	String	作业算子名称。
node_type	String	作业算子类型。
node_qualified_name	String	作业算子Qname。
workspace_id	String	作业算子类型所属空间。

状态码： 400

表 10-355 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-356 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-357 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-358 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-359 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
{
  "total": 2,
  "data": [ {
    "source_guid": "da431e05-804a-491f-945a-a29cfb77e635",
    "source_qualified_name": "wk.dli_01@dli.0833a5737480d53b2f25c010dc1a7b88-workspace-
b88c445407b24283aa949f9833a38fd8",
    "source_type": "dli_table_managed",
    "source_db": "wk",
    "source_schema": null,
    "source_table": "dli_01",
    "target_guid": "f74609f6-8a65-492b-991c-ae39918c69ad",
    "target_qualified_name": "wk.dli_02@dli.0833a5737480d53b2f25c010dc1a7b88-workspace-
b88c445407b24283aa949f9833a38fd8",
    "target_type": "dli_table_managed",
    "target_db": "wk",
    "target_schema": null,
    "target_table": "dli_02",
    "node_guid": "460abeae-aad4-4456-8c35-5ec530506ae2",
    "node_name": "DLI_SQL_6038",
    "node_type": "dli_sql_node",
    "node_qualified_name": "19431260.32e27312-223e-4333-
a231-333122123124@dlf_job.0833a5737480d53b2f25c010dc1a7b88-workspace-
b88c445407b24283aa949f9833a38fd8",
    "workspace_id": "b88c445407b24283aa949f9833a38fd8"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowLineageBulkSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowLineageBulkRequest request = new ShowLineageBulkRequest();
try {
    ShowLineageBulkResponse response = client.showLineageBulk(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowLineageBulkRequest()
        response = client.show_lineage_bulk(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowLineageBulkRequest{}
    response, err := client.ShowLineageBulk(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.10.9 表关联作业算子列表(邀测)

### 功能介绍

查询表相关的作业算子列表，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/datamap/table/{guid}/node

表 10-360 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

### 请求参数

表 10-361 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

### 响应参数

状态码： 200

表 10-362 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of <b>JobAndNodeInfo</b> objects	作业算子基本信息列表。

表 10-363 JobAndNodeInfo

参数	参数类型	描述
task_id	String	作业算子id。
job_name	String	作业算子名称。
workspace_id	String	作业算子所在空间id。

状态码： 400

表 10-364 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-365 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-366 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。



状态码： 404

表 10-367 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-368 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
[ {  
  "task_id" : "937987",  
  "job_name" : "ETL_Job_dli2dws",  
  "workspace_id" : "b88c445407b24283aa949f9833a38fd8"  
}]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowNodesSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowNodesRequest request = new ShowNodesRequest();
    request.withGuid("{guid}");
    try {
        ShowNodesResponse response = client.showNodes(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowNodesRequest()
        request.guid = "{guid}"
        response = client.show_nodes(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowNodesRequest{}
    request.Guid = "{guid}"
    response, err := client.ShowNodes(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。

状态码	描述
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.10.10 产出信息(邀测)

### 功能介绍

查询表相关的作业算子运行实例信息，该接口功能处于邀测阶段，后续将随功能公测将逐步开放。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/datamap/node/{task\_id}/instances

表 10-369 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
task_id	是	String	作业算子id，表关联作业算子列表接口响应体的task_id。

表 10-370 Query 参数

参数	是否必选	参数类型	描述
workspace_id	否	String	空间ID，表关联作业算子列表接口响应体的workspace_id。
job_name	是	String	作业算子名称，表关联作业算子列表接口响应体的job_name。
start_time	是	Number	搜索参数时间范围的开始时间，例：1705248000000。
end_time	是	Number	搜索参数时间范围的结束时间，例：1705311669796
offset	是	Integer	分页参数偏移量。

参数	是否必选	参数类型	描述
limit	是	Integer	分页参每页数量。

## 请求参数

表 10-371 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID</a> 和 <a href="#">工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-372 响应 Body 参数

参数	参数类型	描述
status_code	Integer	响应码。
body	String	响应体。
header_map	Object	响应头信息。

状态码： 400

表 10-373 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-374 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-375 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-376 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-377 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
{  
  "status_code": 200,  
}
```



```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowInstanceInfosRequest request = new ShowInstanceInfosRequest();
request.withTaskId("{task_id}");
try {
    ShowInstanceInfosResponse response = client.showInstanceInfos(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceInfosRequest()
        request.task_id = "{task_id}"
        response = client.show_instance_infos(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```



## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceInfosRequest{}
    request.TaskId = "{task_id}"
    response, err := client.ShowInstanceInfos(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.10.11 批量打标签(邀测)

### 功能介绍

批量给资产打标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/datamap/entities/guids/tags

表 10-378 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 10-379 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
instance	是	String	实例ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-380 请求 Body 参数

参数	是否必选	参数类型	描述
recommendations	否	Array of <a href="#">Recommendation</a> objects	标签信息。
guids	否	Array of strings	资产guid。
add_type	否	String	添加资产类型。cover: 覆盖 追加: append。默认追加。

表 10-381 Recommendation

参数	是否必选	参数类型	描述
name	否	String	标签名称。

## 响应参数

状态码： 400

表 10-382 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-383 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-384 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-385 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 500

表 10-386 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "guids" : [ "3e74c3d7-c5b7-4e7c-827e-40adb1b5ecbf", "90c3b725-27db-45cc-aa88-3dc6eba93d46" ],
  "recommendations" : [ {
    "name" : "MTa"
  }, {
    "name" : "JnB"
  } ],
  "add_type" : "append"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class BatchTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
BatchTagRequest request = new BatchTagRequest();
BatchRecommendationRequest body = new BatchRecommendationRequest();
request.withBody(body);
try {
    BatchTagResponse response = client.batchTag(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchTagRequest()
        request.body = BatchRecommendationRequest(
        )
        response = client.batch_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```

dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchTagRequest{}
    request.Body = &model.BatchRecommendationRequest{
    }
    response, err := client.BatchTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。
500	InternalServerError。

## 10.11 标签接口

## 10.11.1 查询标签列表

### 功能介绍

查询标签列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/tags

表 10-387 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 10-388 Query 参数

参数	是否必选	参数类型	描述
type	否	String	标签类型 缺省值：all。
name	否	String	标签名。
create_user	否	String	标签创建用户。
start	否	String	开始时间。
end	否	String	结束时间。
limit	否	String	分页参数:每页限定数量，缺省值：10。
offset	否	String	分页参数：页数，缺省值：0。
sort_by	否	String	排序字段 默认为createTime，缺省值：createTime。
sort_order	否	String	排序方式 默认排序字段为降序，缺省值：desc。

## 请求参数

表 10-389 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID</a> 和 <a href="#">工作空间ID</a> 。

## 响应参数

状态码： 200

表 10-390 响应 Body 参数

参数	参数类型	描述
count	Integer	总数。
limit	Integer	分页参数limit。
offset	Integer	分页参数offset。
quota	Integer	指标配额。
tags	Array of <a href="#">GlossaryInfo</a> objects	标签信息列表。

表 10-391 GlossaryInfo

参数	参数类型	描述
name	String	标签名称。
description	String	描述。
guid	String	标签的guid。
create_user	String	创建用户。
create_time	Number	创建时间。

状态码： 400



表 10-392 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-393 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-394 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-395 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码： 200

OK。

```
{  
  "count" : 20,  
}
```

```
"limit" : 1,  
"offset" : 1,  
"quota" : 100,  
"tags" : [ {  
  "create_time" : 1663826213551,  
  "create_user" : "user_demo",  
  "description" : null,  
  "guid" : "b4412463-7253-4b60-8184-c938fe46d0aa",  
  "name" : "123"  
} ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowGlossaryListSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowGlossaryListRequest request = new ShowGlossaryListRequest();  
        try {  
            ShowGlossaryListResponse response = client.showGlossaryList(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowGlossaryListRequest()
        response = client.show_glossary_list(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()

request := &model.ShowGlossaryListRequest{}
response, err := client.ShowGlossaryList(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。

## 10.11.2 标签关联到资产

### 功能介绍

标签关联到资产

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/tags/{term\_guid}/assignedentities

表 10-396 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
term_guid	是	String	资产guid，获取方法请参见 <a href="#">数据资产guid</a> 。

## 请求参数

表 10-397 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

表 10-398 请求 Body 参数

参数	是否必选	参数类型	描述
guid	否	String	标签guid。

## 响应参数

状态码： 400

表 10-399 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 401

表 10-400 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 403

表 10-401 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

状态码： 404

表 10-402 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
{
  "guid" : "a2171b94-4e91-4fd9-8ef5-c6dbac66c757"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class AddTagToAssetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
AddTagToAssetRequest request = new AddTagToAssetRequest();
request.withTermGuid("{term_guid}");
CatalogInfo body = new CatalogInfo();
request.withBody(body);
try {
    AddTagToAssetResponse response = client.addTagToAsset(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

## Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = AddTagToAssetRequest()
        request.term_guid = "{term_guid}"
        request.body = CatalogInfo(
            )
        response = client.add_tag_to_asset(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddTagToAssetRequest{}
    request.TermGuid = "{term_guid}"
    request.Body = &model.CatalogInfo{
    }
    response, err := client.AddTagToAsset(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK。
400	BadRequest。
401	Unauthorized。
403	Forbidden。
404	Not Found。



# 11 数据服务 API

## 11.1 API 管理接口

### 11.1.1 创建 API

#### 功能介绍

创建API。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/{project\_id}/service/apis

表 11-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 11-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-3 请求 Body 参数

参数	是否必选	参数类型	描述
catalog_id	否	String	目录ID。
name	否	String	API名称。
description	否	String	API描述。
log_flag	否	Boolean	是否启用访问日志。

参数	是否必选	参数类型	描述
api_type	否	String	API类型。 枚举值： <ul style="list-style-type: none"> <li>API_SPECIFIC_TYPE_CONFIGURATION: 配置类API</li> <li>API_SPECIFIC_TYPE_SCRIPT: 脚本类API</li> <li>API_SPECIFIC_TYPE_MYBATIS: MyBatis类API</li> <li>API_SPECIFIC_TYPE_GROOVY: Groovy类API</li> <li>API_SPECIFIC_TYPE_REGISTRER: 注册类API</li> <li>API_SPECIFIC_TYPE_ORCHESTRATE: 编排类API</li> <li>API_TYPE_CREATE: 创建类API (已弃用)</li> <li>API_TYPE_REGISTER: 注册类API (已弃用)</li> </ul>
auth_type	否	String	认证类型。 枚举值： <ul style="list-style-type: none"> <li>APP: app secret认证</li> <li>IAM: IAM token认证</li> <li>NONE: 无认证</li> </ul>
retention_period	否	Integer	最低保留期限，单位小时。
manager	否	String	API审核人。
path	否	String	API路径。
protocol	否	String	API请求协议类型。 枚举值： <ul style="list-style-type: none"> <li>PROTOCOL_TYPE_HTTP: HTTP协议请求</li> <li>PROTOCOL_TYPE_HTTPS: HTTPS协议请求</li> </ul>
request_type	否	String	请求类型。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_TYPE_POST: POST请求</li> <li>REQUEST_TYPE_GET: GET请求</li> </ul>

参数	是否必选	参数类型	描述
tags	否	Array of strings	标签。
visibility	否	String	API可见性。 枚举值： <ul style="list-style-type: none"> <li>• WORKSPACE: 工作空间可见</li> <li>• PROJECT: 项目可见</li> <li>• DOMAIN: 租户可见</li> </ul>
publish_type	否	String	发布类型(已弃用)。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH_TYPE_PUBLIC: 公开</li> <li>• PUBLISH_TYPE_PRIVATE: 私有</li> </ul>
data_mask_switch	否	Boolean	数据脱敏开关。
data_mask_params	否	Array of <b>DataMaskParams</b> objects	数据脱敏配置。
body	否	String	请求体内容描述，仅注册类API使用。
datasource_config_list	否	Array of <b>DatasourceConfig</b> objects	多数据源配置信息，仅Groovy类API使用。
groovy_content	否	String	Groovy文本，仅Groovy类API使用。
request_params	否	Array of <b>RequestParam</b> objects	API请求参数列表。
datasource_config	否	<b>DatasourceConfig</b> object	API数据源配置。
backend_config	否	<b>BackendConfig</b> object	API后端配置。仅注册类API使用

表 11-4 DataMaskParams

参数	是否必选	参数类型	描述
column_name	否	String	敏感字段。

参数	是否必选	参数类型	描述
algorithm_name	否	String	算法名称。
algorithm_type	否	String	算法类型。
en_name	否	String	算法名称。
algorithm_parameters	否	String	参数。
failure_policy	否	String	失败策略。 枚举值： <ul style="list-style-type: none"> <li>• SKIP: 跳过, 即不做处理</li> <li>• INTERRUPT_AND_EXCEPTION: 中断并抛出异常</li> <li>• SET_NULL: 置空</li> <li>• DEFAULT_VALUE: 默认值</li> </ul>

表 11-5 DatasourceConfig

参数	是否必选	参数类型	描述
type	否	String	数据源的类型。 枚举值： <ul style="list-style-type: none"> <li>• MYSQL: MYSQL数据源</li> <li>• DLI: DLI数据源</li> <li>• DWS: DWS数据源</li> <li>• HIVE: HIVE数据源</li> <li>• HBASE: HBASE数据源</li> </ul>
connection_name	否	String	数据连接名称。
connection_id	否	String	数据连接ID。
database	否	String	数据库名。
datatable	否	String	数据表名称。
table_id	否	String	数据表ID。
queue	否	String	DLI的队列名称。

参数	是否必选	参数类型	描述
access_mode	否	String	获取数据的模式。 枚举值： <ul style="list-style-type: none"> <li>• SQL: SQL取数方式</li> <li>• ROW_KEY: ROW_KEY取数方式 仅适用于HBASE</li> <li>• PREFIX_FILTER: PREFIX_FILTER取数方式仅适用于HBASE</li> </ul>
access_type	否	String	取数方式(已弃用)。 枚举值： <ul style="list-style-type: none"> <li>• SCRIPT: 脚本方式</li> <li>• CONFIGURAITON: 配置方式</li> </ul>
pagination	否	String	分页方式。 枚举值： <ul style="list-style-type: none"> <li>• DEFAULT: 默认分页</li> <li>• CUSTOM: 自定义分页</li> </ul>
sql	否	String	脚本模式下的SQL语句。
backend_params	否	Array of <a href="#">ApiRequestPara</a> objects	API后端参数。
response_params	否	Array of <a href="#">ApiResponsePara</a> objects	配置类API返回参数。
order_paras	否	Array of <a href="#">DatasourceOrderPara</a> objects	排序参数。
total_size_sql	否	String	总条数计算脚本SQL。

表 11-6 RequestPara

参数	是否必选	参数类型	描述
name	否	String	参数名。

参数	是否必选	参数类型	描述
position	否	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_PARAMETER_POSITION_PATH: 请求PATH内参数</li> <li>REQUEST_PARAMETER_POSITION_HEADER: 请求HEADER内参数</li> <li>REQUEST_PARAMETER_POSITION_QUERY: 请求QUERY参数</li> </ul>
type	否	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_PARAMETER_TYPE_NUMBER: 数值类型</li> <li>REQUEST_PARAMETER_TYPE_STRING: 字符串类型</li> </ul>
description	否	String	参数的描述。
necessary	否	Boolean	参数是否必填。
example_value	否	String	示例值。
default_value	否	String	默认值。
support_null	否	Boolean	支持NULL值。

表 11-7 ApiRequestPara

参数	是否必选	参数类型	描述
name	否	String	参数名称。
mapping	否	String	映射字段。

参数	是否必选	参数类型	描述
condition	否	String	操作符。 枚举值： <ul style="list-style-type: none"> <li>• CONDITION_TYPE_EQ: 等于</li> <li>• CONDITION_TYPE_NE: 不等于</li> <li>• CONDITION_TYPE_GT: 大于</li> <li>• CONDITION_TYPE_GE: 大于等于</li> <li>• CONDITION_TYPE_LT: 小于</li> <li>• CONDITION_TYPE_LE: 小于等于</li> <li>• CONDITION_TYPE_LIKE: 模糊查询如:%like%</li> <li>• CONDITION_TYPE_LIKE_L: 模糊查询如:%like</li> <li>• CONDITION_TYPE_LIKE_R: 模糊查询如:like%</li> </ul>

表 11-8 ApiResponsePara

参数	是否必选	参数类型	描述
name	否	String	参数名。
field	否	String	绑定的表字段。
type	否	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_TYPE_NUMBER: 数值类型</li> <li>• REQUEST_PARAMETER_TYPE_STRING: 字符串类型</li> </ul>
description	否	String	参数描述。
example_value	否	String	参数示例值。



表 11-9 DatasourceOrderPara

参数	是否必选	参数类型	描述
name	否	String	排序参数名称。
field	否	String	对应的参数字段。
optional	否	Boolean	是否可选。
sort	否	String	排序方式。 枚举值： <ul style="list-style-type: none"> <li>• ASC: 升序</li> <li>• DESC: 降序</li> <li>• CUSTOM: 自定义</li> </ul>
order	否	Integer	排序参数顺序。
description	否	String	排序参数描述信息。

表 11-10 BackendConfig

参数	是否必选	参数类型	描述
type	否	String	请求类型。 <ul style="list-style-type: none"> <li>• REQUEST_TYPE_POST: POST请求</li> <li>• REQUEST_TYPE_GET: GET请求</li> </ul>
protocol	否	String	API请求协议类型。 <ul style="list-style-type: none"> <li>• PROTOCOL_TYPE_HTTP: HTTP协议请求</li> <li>• PROTOCOL_TYPE_HTTPS: HTTPS协议请求</li> </ul>
host	否	String	后端HOST。
timeout	否	Integer	后端超时时间。
path	否	String	后端请求PATH。
backend_params	否	Array of BackendRequestPara objects	API后端参数。
constant_params	否	Array of BackendConstant objects	后端常量参数。

表 11-11 BackendRequestPara

参数	是否必选	参数类型	描述
name	否	String	API请求参数名称。
position	否	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_POSITION_PATH: 请求PATH内参数</li> <li>• REQUEST_PARAMETER_POSITION_HEADER: 请求HEADER内参数</li> <li>• REQUEST_PARAMETER_POSITION_QUERY: 请求QUERY参数</li> </ul>
backend_para_name	否	String	对应的后端参数。

表 11-12 BackendConstant

参数	是否必选	参数类型	描述
name	否	String	常量参数名。
type	否	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_TYPE_NUMBER: 数值类型</li> <li>• REQUEST_PARAMETER_TYPE_STRING: 字符串类型</li> </ul>
position	否	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_POSITION_PATH: 请求PATH内参数</li> <li>• REQUEST_PARAMETER_POSITION_HEADER: 请求HEADER内参数</li> <li>• REQUEST_PARAMETER_POSITION_QUERY: 请求QUERY参数</li> </ul>
description	否	String	常量参数描述。
value	否	String	常量参数值。

## 响应参数

状态码： 200

表 11-13 响应 Body 参数

参数	参数类型	描述
id	String	创建成功的API ID。

状态码： 400

表 11-14 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

- 创建配置类API。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis
```

```
{
  "catalog_id": "0",
  "name": "create_config_api",
  "description": "dws",
  "visibility": "WORKSPACE",
  "api_type": "API_SPECIFIC_TYPE_CONFIGURATION",
  "auth_type": "APP",
  "retention_period": "1",
  "manager": "user",
  "log_flag": true,
  "path": "/pathrGW6riuO/{p}",
  "protocol": "PROTOCOL_TYPE_HTTPS",
  "request_type": "REQUEST_TYPE_POST",
  "request_paras": [ {
    "name": "a",
    "position": "REQUEST_PARAMETER_POSITION_QUERY",
    "type": "REQUEST_PARAMETER_TYPE_STRING",
    "description": "1",
    "necessary": true,
    "example_value": "1",
    "default_value": "1",
    "support_null": false
  }, {
    "name": "b",
    "position": "REQUEST_PARAMETER_POSITION_HEADER",
    "type": "REQUEST_PARAMETER_TYPE_STRING",
    "description": "1",
    "necessary": true,
    "example_value": "1",
    "default_value": "1",
    "support_null": false
  }, {
    "name": "c",
    "position": "REQUEST_PARAMETER_POSITION_BODY",
```

```

"type": "REQUEST_PARAMETER_TYPE_STRING",
"description": "1",
"necessary": true,
"example_value": "1",
"default_value": "1",
"support_null": false
}, {
"name": "p",
"position": "REQUEST_PARAMETER_POSITION_PATH",
"type": "REQUEST_PARAMETER_TYPE_STRING",
"description": "ok1",
"necessary": true,
"example_value": "ok1",
"default_value": "ok1",
"support_null": false
}],
"datasource_config": {
"access_mode": "SQL",
"order_paras": [],
"table_id": "NativeTable-c68d84bd2d89420cb2458b4e06805ddf-postgres-dbadmin-test3",
"datatable": "dbadmin.test3",
"database": "postgres",
"connection_id": "c68d84bd2d89420cb2458b4e06805ddf",
"connection_name": "yqp_dws",
"type": "DWS",
"pagination": "DEFAULT",
"sql": "",
"total_size_sql": "",
"backend_paras": [ {
"name": "a",
"mapping": "username",
"condition": "CONDITION_TYPE_EQ"
} ],
"response_paras": [ {
"description": "",
"example_value": "",
"field": "id_num",
"name": "id_num",
"type": "REQUEST_PARAMETER_TYPE_STRING"
}, {
"description": "",
"example_value": "",
"field": "username",
"name": "username",
"type": "REQUEST_PARAMETER_TYPE_STRING"
} ]
},
"data_mask_switch": false,
"data_mask_paras": []
}

```

- 创建脚本类API。

```

/v1/0833a5737480d53b2f250010d01a7b88/service/apis
{
"catalog_id": "0",
"name": "create_script_api",
"description": "dws",
"visibility": "WORKSPACE",
"api_type": "API_SPECIFIC_TYPE_SCRIPT",
"auth_type": "APP",
"retention_period": "1",
"manager": "user",
"log_flag": true,
"path": "/pathR7YL3xfN/{p}",
"protocol": "PROTOCOL_TYPE_HTTPS",
"request_type": "REQUEST_TYPE_POST",
"request_paras": [ {
"name": "a",
"position": "REQUEST_PARAMETER_POSITION_QUERY",

```

```

"type": "REQUEST_PARAMETER_TYPE_STRING",
"description": "1",
"necessary": true,
"example_value": "1",
"default_value": "1",
"support_null": false
}, {
"name": "b",
"position": "REQUEST_PARAMETER_POSITION_HEADER",
"type": "REQUEST_PARAMETER_TYPE_STRING",
"description": "1",
"necessary": true,
"example_value": "1",
"default_value": "1",
"support_null": false
}, {
"name": "c",
"position": "REQUEST_PARAMETER_POSITION_BODY",
"type": "REQUEST_PARAMETER_TYPE_STRING",
"description": "1",
"necessary": true,
"example_value": "1",
"default_value": "1",
"support_null": false
}, {
"name": "p",
"position": "REQUEST_PARAMETER_POSITION_PATH",
"type": "REQUEST_PARAMETER_TYPE_STRING",
"description": "ok1",
"necessary": true,
"example_value": "ok1",
"default_value": "ok1",
"support_null": false
}],
"datasource_config": {
"access_mode": "SQL",
"order_paras": [],
"database": "dls",
"connection_id": "c68d84bd2d89420cb2458b4e06805ddf",
"connection_name": "yqp_dws",
"type": "DWS",
"pagination": "CUSTOM",
"sql": "c2VsZWN0IDMsNCw1",
"total_size_sql": "",
"backend_paras": [ {
"name": "a",
"mapping": "a",
"condition": "CONDITION_TYPE_EQ"
}, {
"name": "b",
"mapping": "b",
"condition": "CONDITION_TYPE_EQ"
}, {
"name": "c",
"mapping": "c",
"condition": "CONDITION_TYPE_EQ"
}, {
"name": "p",
"mapping": "p",
"condition": "CONDITION_TYPE_EQ"
} ]
},
"data_mask_switch": false,
"data_mask_paras": []
}

```

- 创建Mybatis类API。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis
```

```
{
```

```

"catalog_id" : "0",
"name" : "create_mybatis_api",
"description" : "desc",
"visibility" : "WORKSPACE",
"api_type" : "API_SPECIFIC_TYPE_MYBATIS",
"auth_type" : "APP",
"retention_period" : "0",
"manager" : "user",
"log_flag" : true,
"path" : "/pathR7YL3xfN",
"protocol" : "PROTOCOL_TYPE_HTTPS",
"request_type" : "REQUEST_TYPE_POST",
"request_paras" : [ {
  "name" : "ssss",
  "position" : "REQUEST_PARAMETER_POSITION_QUERY",
  "type" : "REQUEST_PARAMETER_TYPE_STRING",
  "description" : "1",
  "necessary" : true,
  "example_value" : "1",
  "default_value" : "1",
  "support_null" : false
}],
"datasource_config" : {
  "access_mode" : "SQL",
  "order_paras" : [ ],
  "database" : "dls",
  "connection_id" : "c68d84bd2d89420cb2458b4e06805ddf",
  "connection_name" : "yqp_dws",
  "type" : "DWS",
  "pagination" : "CUSTOM",
  "sql" :
"c2VsZWN0IDEKPGlmIHJlc3Q9InNzc3MgPT0gJzEnLnRvU3RyaW5nKCKiPiAgCiwylAo8L2lmPg==",
  "total_size_sql" : "",
  "backend_paras" : [ ]
},
"data_mask_switch" : false,
"data_mask_paras" : [ ]
}

```

## 响应示例

**状态码：** 200

Success。

```

{
  "id" : "6b9e682fd6d0ef7c0f67470124b20762"
}

```

## SDK 代码示例

SDK代码示例如下。

### Java

- 创建配置类API。

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

```

```
import java.util.List;
import java.util.ArrayList;

public class CreateApiSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateApiRequest request = new CreateApiRequest();
        Api body = new Api();
        List<ApiResponsePara> listDatasourceConfigResponseParas = new ArrayList<>();
        listDatasourceConfigResponseParas.add(
            new ApiResponsePara()
                .withName("id_num")
                .withField("id_num")
                .withType(ApiResponsePara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
                .withDescription("")
                .withExampleValue("")
        );
        listDatasourceConfigResponseParas.add(
            new ApiResponsePara()
                .withName("username")
                .withField("username")
                .withType(ApiResponsePara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
                .withDescription("")
                .withExampleValue("")
        );
        List<ApiRequestPara> listDatasourceConfigBackendParas = new ArrayList<>();
        listDatasourceConfigBackendParas.add(
            new ApiRequestPara()
                .withName("a")
                .withMapping("username")
                .withCondition(ApiRequestPara.ConditionEnum.fromValue("CONDITION_TYPE_EQ"))
        );
        DatasourceConfig datasourceConfigbody = new DatasourceConfig();
        datasourceConfigbody.withType(DatasourceConfig.TypeEnum.fromValue("DWS"))
            .withConnectionName("yqp_dws")
            .withConnectionId("c68d84bd2d89420cb2458b4e06805ddf")
            .withDatabase("postgres")
            .withDatatable("dbadmin.test3")
            .withTableId("NativeTable-c68d84bd2d89420cb2458b4e06805ddf-postgres-dbadmin-test3")
            .withAccessMode(DatasourceConfig.AccessModeEnum.fromValue("SQL"))
            .withPagination(DatasourceConfig.PaginationEnum.fromValue("DEFAULT"))
            .withSql("")
            .withBackendParas(listDatasourceConfigBackendParas)
            .withResponseParas(listDatasourceConfigResponseParas);
        List<RequestPara> listbodyRequestParas = new ArrayList<>();
        listbodyRequestParas.add(
            new RequestPara()
                .withName("a")
                .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_QU
```

```

ERY"))
        .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
        .withDescription("1")
        .withNecessary(true)
        .withExampleValue("1")
        .withDefaultValue("1")
    );
    listbodyRequestParas.add(
        new RequestPara()
            .withName("b")
            .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_HE
ADER"))
            .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
            .withDescription("1")
            .withNecessary(true)
            .withExampleValue("1")
            .withDefaultValue("1")
    );
    listbodyRequestParas.add(
        new RequestPara()
            .withName("c")
            .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_BO
DY"))
            .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
            .withDescription("1")
            .withNecessary(true)
            .withExampleValue("1")
            .withDefaultValue("1")
    );
    listbodyRequestParas.add(
        new RequestPara()
            .withName("p")
            .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_PAT
H"))
            .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
            .withDescription("ok1")
            .withNecessary(true)
            .withExampleValue("ok1")
            .withDefaultValue("ok1")
    );
    body.withDatasourceConfig(datasourceConfigbody);
    body.withRequestParas(listbodyRequestParas);
    body.withVisibility(Api.VisibilityEnum.fromValue("WORKSPACE"));
    body.withRequestType(Api.RequestTypeEnum.fromValue("REQUEST_TYPE_POST"));
    body.withProtocol(Api.ProtocolEnum.fromValue("PROTOCOL_TYPE_HTTPS"));
    body.withPath("/pathrGW6riuO/{p}");
    body.withManager("user");
    body.withAuthType(Api.AuthTypeEnum.fromValue("APP"));
    body.withApiType(Api.ApiTypeEnum.fromValue("API_SPECIFIC_TYPE_CONFIGURATION"));
    body.withLogFlag(true);
    body.withDescription("dws");
    body.withName("create_config_api");
    body.withCatalogId("0");
    request.withBody(body);
    try {
        CreateApiResponse response = client.createApi(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}

```



```
}
}
```

- 创建脚本类API。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateApiSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateApiRequest request = new CreateApiRequest();
        Api body = new Api();
        List<ApiRequestPara> listDatasourceConfigBackendParas = new ArrayList<>();
        listDatasourceConfigBackendParas.add(
            new ApiRequestPara()
                .withName("a")
                .withMapping("a")
                .withCondition(ApiRequestPara.ConditionEnum.fromValue("CONDITION_TYPE_EQ"))
        );
        listDatasourceConfigBackendParas.add(
            new ApiRequestPara()
                .withName("b")
                .withMapping("b")
                .withCondition(ApiRequestPara.ConditionEnum.fromValue("CONDITION_TYPE_EQ"))
        );
        listDatasourceConfigBackendParas.add(
            new ApiRequestPara()
                .withName("c")
                .withMapping("c")
                .withCondition(ApiRequestPara.ConditionEnum.fromValue("CONDITION_TYPE_EQ"))
        );
        listDatasourceConfigBackendParas.add(
            new ApiRequestPara()
                .withName("p")
                .withMapping("p")
                .withCondition(ApiRequestPara.ConditionEnum.fromValue("CONDITION_TYPE_EQ"))
        );
        DatasourceConfig datasourceConfigbody = new DatasourceConfig();
        datasourceConfigbody.withType(DatasourceConfig.TypeEnum.fromValue("DWS"))
            .withConnectionName("yqp_dws")
```

```

        .withConnectionId("c68d84bd2d89420cb2458b4e06805ddf")
        .withDatabase("dls")
        .withAccessMode(DatasourceConfig.AccessModeEnum.fromValue("SQL"))
        .withPagination(DatasourceConfig.PaginationEnum.fromValue("CUSTOM"))
        .withSql("c2VsZWN0IDMsNCw1")
        .withBackendParas(listDatasourceConfigBackendParas);
List<RequestPara> listbodyRequestParas = new ArrayList<>();
listbodyRequestParas.add(
    new RequestPara()
        .withName("a")
        .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_QUE
ERY"))
        .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
        .withDescription("1")
        .withNecessary(true)
        .withExampleValue("1")
        .withDefaultValue("1")
    );
listbodyRequestParas.add(
    new RequestPara()
        .withName("b")
        .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_HE
ADER"))
        .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
        .withDescription("1")
        .withNecessary(true)
        .withExampleValue("1")
        .withDefaultValue("1")
    );
listbodyRequestParas.add(
    new RequestPara()
        .withName("c")
        .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_BO
DY"))
        .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
        .withDescription("1")
        .withNecessary(true)
        .withExampleValue("1")
        .withDefaultValue("1")
    );
listbodyRequestParas.add(
    new RequestPara()
        .withName("p")
        .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_PAT
H"))
        .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
        .withDescription("ok1")
        .withNecessary(true)
        .withExampleValue("ok1")
        .withDefaultValue("ok1")
    );
body.withDatasourceConfig(datasourceConfigbody);
body.withRequestParas(listbodyRequestParas);
body.withVisibility(Api.VisibilityEnum.fromValue("WORKSPACE"));
body.withRequestType(Api.RequestTypeEnum.fromValue("REQUEST_TYPE_POST"));
body.withProtocol(Api.ProtocolEnum.fromValue("PROTOCOL_TYPE_HTTPS"));
body.withPath("/pathR7YL3xfN/{p}");
body.withManager("user");
body.withAuthType(Api.AuthTypeEnum.fromValue("APP"));
body.withApiType(Api.ApiTypeEnum.fromValue("API_SPECIFIC_TYPE_SCRIPT"));
body.withLogFlag(true);
body.withDescription("dws");
body.withName("create_script_api");
body.withCatalogId("0");
request.withBody(body);
try {
    CreateApiResponse response = client.createApi(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {

```

```

        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
}

```

- 创建Mybatis类API。

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateApiSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateApiRequest request = new CreateApiRequest();
        Api body = new Api();
        DatasourceConfig datasourceConfigbody = new DatasourceConfig();
        datasourceConfigbody.withType(DatasourceConfig.TypeEnum.fromValue("DWS"))
            .withConnectionName("yqp_dws")
            .withConnectionId("c68d84bd2d89420cb2458b4e06805ddf")
            .withDatabase("dls")
            .withAccessMode(DatasourceConfig.AccessModeEnum.fromValue("SQL"))
            .withPagination(DatasourceConfig.PaginationEnum.fromValue("CUSTOM"))
            .withSql("c2VsZWN0IDEKPGImIHRLc3Q9InNzc3MgPT0gJzEnLnRvU3RyaW5nKCKiPiAgCiwylAo8L
2lmPg==");
        List<RequestPara> listbodyRequestParas = new ArrayList<>();
        listbodyRequestParas.add(
            new RequestPara()
                .withName("ssss")
                .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_QU
ERY"))
                .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_STRING"))
                .withDescription("1")
                .withNecessary(true)

```

```
        .withExampleValue("1")
        .withDefaultValue("1")
    );
    body.withDatasourceConfig(datasourceConfigbody);
    body.withRequestParas(listbodyRequestParas);
    body.withVisibility(Api.VisibilityEnum.fromValue("WORKSPACE"));
    body.withRequestType(Api.RequestTypeEnum.fromValue("REQUEST_TYPE_POST"));
    body.withProtocol(Api.ProtocolEnum.fromValue("PROTOCOL_TYPE_HTTPS"));
    body.withPath("/pathR7YL3xfN");
    body.withManager("user");
    body.withAuthType(Api.AuthTypeEnum.fromValue("APP"));
    body.withApiType(Api.ApiTypeEnum.fromValue("API_SPECIFIC_TYPE_MYBATIS"));
    body.withLogFlag(true);
    body.withDescription("desc");
    body.withName("create_mybatis_api");
    body.withCatalogId("0");
    request.withBody(body);
    try {
        CreateApiResponse response = client.createApi(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
    }
}
```

## Python

- 创建配置类API。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateApiRequest()
        listResponseParasDatasourceConfig = [
            ApiResponsePara(
                name="id_num",
                field="id_num",
            )
        ]
```

```
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="",
        example_value=""
    ),
    ApiResponsePara(
        name="username",
        field="username",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="",
        example_value=""
    )
]
listBackendParasDatasourceConfig = [
    ApiRequestPara(
        name="a",
        mapping="username",
        condition="CONDITION_TYPE_EQ"
    )
]
datasourceConfigbody = DatasourceConfig(
    type="DWS",
    connection_name="yqp_dws",
    connection_id="c68d84bd2d89420cb2458b4e06805ddf",
    database="postgres",
    datatable="dbadmin.test3",
    table_id="NativeTable-c68d84bd2d89420cb2458b4e06805ddf-postgres-dbadmin-test3",
    access_mode="SQL",
    pagination="DEFAULT",
    sql="",
    backend_paras=listBackendParasDatasourceConfig,
    response_paras=listResponseParasDatasourceConfig
)
listRequestParasbody = [
    RequestPara(
        name="a",
        position="REQUEST_PARAMETER_POSITION_QUERY",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="1",
        necessary=True,
        example_value="1",
        default_value="1"
    ),
    RequestPara(
        name="b",
        position="REQUEST_PARAMETER_POSITION_HEADER",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="1",
        necessary=True,
        example_value="1",
        default_value="1"
    ),
    RequestPara(
        name="c",
        position="REQUEST_PARAMETER_POSITION_BODY",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="1",
        necessary=True,
        example_value="1",
        default_value="1"
    ),
    RequestPara(
        name="p",
        position="REQUEST_PARAMETER_POSITION_PATH",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="ok1",
        necessary=True,
        example_value="ok1",
        default_value="ok1"
    )
]
```

```

]
request.body = Api(
  datasource_config=datasourceConfigbody,
  request_paras=listRequestParasbody,
  visibility="WORKSPACE",
  request_type="REQUEST_TYPE_POST",
  protocol="PROTOCOL_TYPE_HTTPS",
  path="/pathrGW6riuO/{p}",
  manager="user",
  auth_type="APP",
  api_type="API_SPECIFIC_TYPE_CONFIGURATION",
  log_flag=True,
  description="dws",
  name="create_config_api",
  catalog_id="0"
)
response = client.create_api(request)
print(response)
except exceptions.ClientRequestException as e:
  print(e.status_code)
  print(e.request_id)
  print(e.error_code)
  print(e.error_msg)

```

- 创建脚本类API。

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateApiRequest()
        listBackendParasDatasourceConfig = [
            ApiRequestPara(
                name="a",
                mapping="a",
                condition="CONDITION_TYPE_EQ"
            ),
            ApiRequestPara(
                name="b",
                mapping="b",
                condition="CONDITION_TYPE_EQ"
            ),
            ApiRequestPara(
                name="c",
                mapping="c",
                condition="CONDITION_TYPE_EQ"
            ),
            ApiRequestPara(

```

```
        name="p",
        mapping="p",
        condition="CONDITION_TYPE_EQ"
    )
]
datasourceConfigbody = DatasourceConfig(
    type="DWS",
    connection_name="yqp_dws",
    connection_id="c68d84bd2d89420cb2458b4e06805ddf",
    database="dls",
    access_mode="SQL",
    pagination="CUSTOM",
    sql="c2VsZWN0IDMsNCw1",
    backend_paras=listBackendParasDatasourceConfig
)
listRequestParasbody = [
    RequestPara(
        name="a",
        position="REQUEST_PARAMETER_POSITION_QUERY",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="1",
        necessary=True,
        example_value="1",
        default_value="1"
    ),
    RequestPara(
        name="b",
        position="REQUEST_PARAMETER_POSITION_HEADER",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="1",
        necessary=True,
        example_value="1",
        default_value="1"
    ),
    RequestPara(
        name="c",
        position="REQUEST_PARAMETER_POSITION_BODY",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="1",
        necessary=True,
        example_value="1",
        default_value="1"
    ),
    RequestPara(
        name="p",
        position="REQUEST_PARAMETER_POSITION_PATH",
        type="REQUEST_PARAMETER_TYPE_STRING",
        description="ok1",
        necessary=True,
        example_value="ok1",
        default_value="ok1"
    )
]
request.body = Api(
    datasource_config=datasourceConfigbody,
    request_paras=listRequestParasbody,
    visibility="WORKSPACE",
    request_type="REQUEST_TYPE_POST",
    protocol="PROTOCOL_TYPE_HTTPS",
    path="/pathR7YL3xfN/{p}",
    manager="user",
    auth_type="APP",
    api_type="API_SPECIFIC_TYPE_SCRIPT",
    log_flag=True,
    description="dws",
    name="create_script_api",
    catalog_id="0"
)
response = client.create_api(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 创建Mybatis类API。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateApiRequest()
        datasourceConfigbody = DatasourceConfig(
            type="DWS",
            connection_name="yqp_dws",
            connection_id="c68d84bd2d89420cb2458b4e06805ddf",
            database="dls",
            access_mode="SQL",
            pagination="CUSTOM",

sql="c2VsZWN0IDEKPGlmIHRLc3Q9InNzc3MgPT0gJzEnLnRvU3RyaW5nKCKiPiAgCiwylAo8L2lmPg=="
        )
        listRequestParasbody = [
            RequestPara(
                name="ssss",
                position="REQUEST_PARAMETER_POSITION_QUERY",
                type="REQUEST_PARAMETER_TYPE_STRING",
                description="1",
                necessary=True,
                example_value="1",
                default_value="1"
            )
        ]
        request.body = Api(
            datasource_config=datasourceConfigbody,
            request_paras=listRequestParasbody,
            visibility="WORKSPACE",
            request_type="REQUEST_TYPE_POST",
            protocol="PROTOCOL_TYPE_HTTPS",
            path="/pathR7YL3xfN",
            manager="user",
            auth_type="APP",
            api_type="API_SPECIFIC_TYPE_MYBATIS",
            log_flag=True,
            description="desc",
            name="create_mybatis_api",
```



```
        catalog_id="0"
    )
    response = client.create_api(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

- 创建配置类API。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateApiRequest{}
    nameResponseParas:= "id_num"
    fieldResponseParas:= "id_num"
    typeResponseParas:= model.GetApiResponseParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
    descriptionResponseParas:= ""
    exampleValueResponseParas:= ""
    nameResponseParas1:= "username"
    fieldResponseParas1:= "username"
    typeResponseParas1:= model.GetApiResponseParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
    descriptionResponseParas1:= ""
    exampleValueResponseParas1:= ""
    var listResponseParasDatasourceConfig = []model.ApiResponsePara{
        {
            Name: &nameResponseParas,
            Field: &fieldResponseParas,
            Type: &typeResponseParas,
            Description: &descriptionResponseParas,
            ExampleValue: &exampleValueResponseParas,
        },
        {
            Name: &nameResponseParas1,
            Field: &fieldResponseParas1,
            Type: &typeResponseParas1,
        }
    }
}
```

```

        Description: &descriptionResponseParas1,
        ExampleValue: &exampleValueResponseParas1,
    },
}
nameBackendParas:= "a"
mappingBackendParas:= "username"
conditionBackendParas:= model.GetApiRequestParaConditionEnum().CONDITION_TYPE_EQ
var listBackendParasDatasourceConfig = []model.ApiRequestPara{
    {
        Name: &nameBackendParas,
        Mapping: &mappingBackendParas,
        Condition: &conditionBackendParas,
    },
}
typeDatasourceConfig:= model.GetDatasourceConfigTypeEnum().DWS
connectionNameDatasourceConfig:= "yqp_dws"
connectionIdDatasourceConfig:= "c68d84bd2d89420cb2458b4e06805ddf"
databaseDatasourceConfig:= "postgres"
datatableDatasourceConfig:= "dbadmin.test3"
tableIdDatasourceConfig:= "NativeTable-c68d84bd2d89420cb2458b4e06805ddf-postgres-dbadmin-
test3"
accessModeDatasourceConfig:= model.GetDatasourceConfigAccessModeEnum().SQL
paginationDatasourceConfig:= model.GetDatasourceConfigPaginationEnum().DEFAULT
sqlDatasourceConfig:= ""
datasourceConfigbody := &model.DatasourceConfig{
    Type: &typeDatasourceConfig,
    ConnectionName: &connectionNameDatasourceConfig,
    ConnectionId: &connectionIdDatasourceConfig,
    Database: &databaseDatasourceConfig,
    Datatable: &datatableDatasourceConfig,
    TableId: &tableIdDatasourceConfig,
    AccessMode: &accessModeDatasourceConfig,
    Pagination: &paginationDatasourceConfig,
    Sql: &sqlDatasourceConfig,
    BackendParas: &listBackendParasDatasourceConfig,
    ResponseParas: &listResponseParasDatasourceConfig,
}
nameRequestParas:= "a"
positionRequestParas:=
model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_QUERY
typeRequestParas:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
descriptionRequestParas:= "1"
necessaryRequestParas:= true
exampleValueRequestParas:= "1"
defaultValueRequestParas:= "1"
nameRequestParas1:= "b"
positionRequestParas1:=
model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_HEADER
typeRequestParas1:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
descriptionRequestParas1:= "1"
necessaryRequestParas1:= true
exampleValueRequestParas1:= "1"
defaultValueRequestParas1:= "1"
nameRequestParas2:= "c"
positionRequestParas2:=
model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_BODY
typeRequestParas2:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
descriptionRequestParas2:= "1"
necessaryRequestParas2:= true
exampleValueRequestParas2:= "1"
defaultValueRequestParas2:= "1"
nameRequestParas3:= "p"
positionRequestParas3:=
model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_PATH
typeRequestParas3:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
descriptionRequestParas3:= "ok1"
necessaryRequestParas3:= true
exampleValueRequestParas3:= "ok1"
defaultValueRequestParas3:= "ok1"

```

```

var listRequestParasbody = []model.RequestPara{
    {
        Name: &nameRequestParas,
        Position: &positionRequestParas,
        Type: &typeRequestParas,
        Description: &descriptionRequestParas,
        Necessary: &necessaryRequestParas,
        ExampleValue: &exampleValueRequestParas,
        DefaultValue: &defaultValueRequestParas,
    },
    {
        Name: &nameRequestParas1,
        Position: &positionRequestParas1,
        Type: &typeRequestParas1,
        Description: &descriptionRequestParas1,
        Necessary: &necessaryRequestParas1,
        ExampleValue: &exampleValueRequestParas1,
        DefaultValue: &defaultValueRequestParas1,
    },
    {
        Name: &nameRequestParas2,
        Position: &positionRequestParas2,
        Type: &typeRequestParas2,
        Description: &descriptionRequestParas2,
        Necessary: &necessaryRequestParas2,
        ExampleValue: &exampleValueRequestParas2,
        DefaultValue: &defaultValueRequestParas2,
    },
    {
        Name: &nameRequestParas3,
        Position: &positionRequestParas3,
        Type: &typeRequestParas3,
        Description: &descriptionRequestParas3,
        Necessary: &necessaryRequestParas3,
        ExampleValue: &exampleValueRequestParas3,
        DefaultValue: &defaultValueRequestParas3,
    },
}
visibilityApi:= model.GetApiVisibilityEnum().WORKSPACE
requestTypeApi:= model.GetApiRequestTypeEnum().REQUEST_TYPE_POST
protocolApi:= model.GetApiProtocolEnum().PROTOCOL_TYPE_HTTPS
pathApi:= "/pathrGW6riuO/{p}"
managerApi:= "user"
authTypeApi:= model.GetApiAuthTypeEnum().APP
apiTypeApi:= model.GetApiApiTypeEnum().API_SPECIFIC_TYPE_CONFIGURATION
logFlagApi:= true
descriptionApi:= "dws"
nameApi:= "create_config_api"
catalogIdApi:= "0"
request.Body = &model.Api{
    DatasourceConfig: datasourceConfigbody,
    RequestParas: &listRequestParasbody,
    Visibility: &visibilityApi,
    RequestType: &requestTypeApi,
    Protocol: &protocolApi,
    Path: &pathApi,
    Manager: &managerApi,
    AuthType: &authTypeApi,
    ApiType: &apiTypeApi,
    LogFlag: &logFlagApi,
    Description: &descriptionApi,
    Name: &nameApi,
    CatalogId: &catalogIdApi,
}
response, err := client.CreateApi(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}

```

- 创建脚本类API。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateApiRequest{}
    nameBackendParas:= "a"
    mappingBackendParas:= "a"
    conditionBackendParas:= model.GetApiRequestParaConditionEnum().CONDITION_TYPE_EQ
    nameBackendParas1:= "b"
    mappingBackendParas1:= "b"
    conditionBackendParas1:= model.GetApiRequestParaConditionEnum().CONDITION_TYPE_EQ
    nameBackendParas2:= "c"
    mappingBackendParas2:= "c"
    conditionBackendParas2:= model.GetApiRequestParaConditionEnum().CONDITION_TYPE_EQ
    nameBackendParas3:= "p"
    mappingBackendParas3:= "p"
    conditionBackendParas3:= model.GetApiRequestParaConditionEnum().CONDITION_TYPE_EQ
    var listBackendParasDatasourceConfig = []model.ApiRequestPara{
        {
            Name: &nameBackendParas,
            Mapping: &mappingBackendParas,
            Condition: &conditionBackendParas,
        },
        {
            Name: &nameBackendParas1,
            Mapping: &mappingBackendParas1,
            Condition: &conditionBackendParas1,
        },
        {
            Name: &nameBackendParas2,
            Mapping: &mappingBackendParas2,
            Condition: &conditionBackendParas2,
        },
        {
            Name: &nameBackendParas3,
            Mapping: &mappingBackendParas3,
            Condition: &conditionBackendParas3,
        }
    }
}
```

```

    },
  }
  typeDatasourceConfig:= model.GetDatasourceConfigTypeEnum().DWS
  connectionNameDatasourceConfig:= "yqp_dws"
  connectionIdDatasourceConfig:= "c68d84bd2d89420cb2458b4e06805ddf"
  databaseDatasourceConfig:= "dls"
  accessModeDatasourceConfig:= model.GetDatasourceConfigAccessModeEnum().SQL
  paginationDatasourceConfig:= model.GetDatasourceConfigPaginationEnum().CUSTOM
  sqlDatasourceConfig:= "c2VsZWNOIDMsNCw1"
  datasourceConfigbody := &model.DatasourceConfig{
    Type: &typeDatasourceConfig,
    ConnectionName: &connectionNameDatasourceConfig,
    ConnectionId: &connectionIdDatasourceConfig,
    Database: &databaseDatasourceConfig,
    AccessMode: &accessModeDatasourceConfig,
    Pagination: &paginationDatasourceConfig,
    Sql: &sqlDatasourceConfig,
    BackendParas: &listBackendParasDatasourceConfig,
  }
  nameRequestParas:= "a"
  positionRequestParas:=
  model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_QUERY
  typeRequestParas:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
  descriptionRequestParas:= "1"
  necessaryRequestParas:= true
  exampleValueRequestParas:= "1"
  defaultValueRequestParas:= "1"
  nameRequestParas1:= "b"
  positionRequestParas1:=
  model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_HEADER
  typeRequestParas1:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
  descriptionRequestParas1:= "1"
  necessaryRequestParas1:= true
  exampleValueRequestParas1:= "1"
  defaultValueRequestParas1:= "1"
  nameRequestParas2:= "c"
  positionRequestParas2:=
  model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_BODY
  typeRequestParas2:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
  descriptionRequestParas2:= "1"
  necessaryRequestParas2:= true
  exampleValueRequestParas2:= "1"
  defaultValueRequestParas2:= "1"
  nameRequestParas3:= "p"
  positionRequestParas3:=
  model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_PATH
  typeRequestParas3:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
  descriptionRequestParas3:= "ok1"
  necessaryRequestParas3:= true
  exampleValueRequestParas3:= "ok1"
  defaultValueRequestParas3:= "ok1"
  var listRequestParasbody = []model.RequestPara{
    {
      Name: &nameRequestParas,
      Position: &positionRequestParas,
      Type: &typeRequestParas,
      Description: &descriptionRequestParas,
      Necessary: &necessaryRequestParas,
      ExampleValue: &exampleValueRequestParas,
      DefaultValue: &defaultValueRequestParas,
    },
  },
  {
    Name: &nameRequestParas1,
    Position: &positionRequestParas1,
    Type: &typeRequestParas1,
    Description: &descriptionRequestParas1,
    Necessary: &necessaryRequestParas1,
    ExampleValue: &exampleValueRequestParas1,
    DefaultValue: &defaultValueRequestParas1,
  },
}

```

```

    },
    {
      Name: &nameRequestParas2,
      Position: &positionRequestParas2,
      Type: &typeRequestParas2,
      Description: &descriptionRequestParas2,
      Necessary: &necessaryRequestParas2,
      ExampleValue: &exampleValueRequestParas2,
      DefaultValue: &defaultValueRequestParas2,
    },
    {
      Name: &nameRequestParas3,
      Position: &positionRequestParas3,
      Type: &typeRequestParas3,
      Description: &descriptionRequestParas3,
      Necessary: &necessaryRequestParas3,
      ExampleValue: &exampleValueRequestParas3,
      DefaultValue: &defaultValueRequestParas3,
    },
  },
}
visibilityApi:= model.GetApiVisibilityEnum().WORKSPACE
requestTypeApi:= model.GetApiRequestTypeEnum().REQUEST_TYPE_POST
protocolApi:= model.GetApiProtocolEnum().PROTOCOL_TYPE_HTTPS
pathApi:= "/pathR7YL3xfN/{p}"
managerApi:= "user"
authTypeApi:= model.GetApiAuthTypeEnum().APP
apiTypeApi:= model.GetApiApiTypeEnum().API_SPECIFIC_TYPE_SCRIPT
logFlagApi:= true
descriptionApi:= "dws"
nameApi:= "create_script_api"
catalogIdApi:= "0"
request.Body = &model.Api{
  DatasourceConfig: datasourceConfigbody,
  RequestParas: &listRequestParasbody,
  Visibility: &visibilityApi,
  RequestType: &requestTypeApi,
  Protocol: &protocolApi,
  Path: &pathApi,
  Manager: &managerApi,
  AuthType: &authTypeApi,
  ApiType: &apiTypeApi,
  LogFlag: &logFlagApi,
  Description: &descriptionApi,
  Name: &nameApi,
  CatalogId: &catalogIdApi,
}
response, err := client.CreateApi(request)
if err == nil {
  fmt.Printf("%+v\n", response)
} else {
  fmt.Println(err)
}
}
}

```

- 创建Mybatis类API。

```

package main

import (
  "fmt"
  "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
  dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
  "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
  region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
  // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
  // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
  // environment variables and decrypted during use to ensure security.
  // In this example, AK and SK are stored in environment variables for authentication. Before

```

running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```

ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateApiRequest{}
typeDatasourceConfig:= model.GetDatasourceConfigTypeEnum().DWS
connectionNameDatasourceConfig:= "yqp_dws"
connectionIdDatasourceConfig:= "c68d84bd2d89420cb2458b4e06805ddf"
databaseDatasourceConfig:= "dls"
accessModeDatasourceConfig:= model.GetDatasourceConfigAccessModeEnum().SQL
paginationDatasourceConfig:= model.GetDatasourceConfigPaginationEnum().CUSTOM
sqlDatasourceConfig:=
"c2VsZWNO1DEKPGlmHRlc3Q9InNzc3MgPT0gJzEnLnRvU3RyaW5nKCKiPiAgCiwylAo8L2lmPg=="
datasourceConfigbody := &model.DatasourceConfig{
    Type: &typeDatasourceConfig,
    ConnectionName: &connectionNameDatasourceConfig,
    ConnectionId: &connectionIdDatasourceConfig,
    Database: &databaseDatasourceConfig,
    AccessMode: &accessModeDatasourceConfig,
    Pagination: &paginationDatasourceConfig,
    Sql: &sqlDatasourceConfig,
}
nameRequestParas:= "ssss"
positionRequestParas:=
model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_QUERY
typeRequestParas:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_STRING
descriptionRequestParas:= "1"
necessaryRequestParas:= true
exampleValueRequestParas:= "1"
defaultValueRequestParas:= "1"
var listRequestParasbody = []model.RequestPara{
    {
        Name: &nameRequestParas,
        Position: &positionRequestParas,
        Type: &typeRequestParas,
        Description: &descriptionRequestParas,
        Necessary: &necessaryRequestParas,
        ExampleValue: &exampleValueRequestParas,
        DefaultValue: &defaultValueRequestParas,
    },
}
visibilityApi:= model.GetApiVisibilityEnum().WORKSPACE
requestTypeApi:= model.GetApiRequestTypeEnum().REQUEST_TYPE_POST
protocolApi:= model.GetApiProtocolEnum().PROTOCOL_TYPE_HTTPS
pathApi:= "/pathR7YL3xfN"
managerApi:= "user"
authTypeApi:= model.GetApiAuthTypeEnum().APP
apiTypeApi:= model.GetApiApiTypeEnum().API_SPECIFIC_TYPE_MYBATIS
logFlagApi:= true
descriptionApi:= "desc"
nameApi:= "create_mybatis_api"
catalogIdApi:= "0"
request.Body = &model.Api{
    DatasourceConfig: datasourceConfigbody,
    RequestParas: &listRequestParasbody,
}

```

```

Visibility: &visibilityApi,
RequestType: &requestTypeApi,
Protocol: &protocolApi,
Path: &pathApi,
Manager: &managerApi,
AuthType: &authTypeApi,
ApiType: &apiTypeApi,
LogFlag: &logFlagApi,
Description: &descriptionApi,
Name: &nameApi,
CatalogId: &catalogIdApi,
}
response, err := client.CreateApi(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.1.2 查询 API 列表

### 功能介绍

查询API列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/apis

表 11-15 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。



表 11-16 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或limit的整数倍, 不满足则向下取整。
limit	否	Integer	查询条数限制。
name	否	String	根据API名称模糊查询。
description	否	String	根据API描述信息模糊查询。
create_user	否	String	根据API创建用户模糊查询。
start_time	否	String	根据API创建时间过滤, 开始时间, 格式遵循RFC3339, 精确到秒, UTC时区, 如 2024-02-24T16:00:00.000Z。
end_time	否	String	根据API创建时间过滤, 结束时间, 格式遵循RFC3339, 精确到秒, UTC时区, 如 2024-04-05T15:59:59.998Z。
tags	否	Array of strings	标签。
api_type	否	String	API类型。 枚举值: <ul style="list-style-type: none"> <li>• API_SPECIFIC_TYPE_CONFIGURATION: 配置类API</li> <li>• API_SPECIFIC_TYPE_SCRIPT: 脚本类API</li> <li>• API_SPECIFIC_TYPE_MYBATIS: MyBatis类API</li> <li>• API_SPECIFIC_TYPE_GROOVY: Groovy类API</li> <li>• API_SPECIFIC_TYPE_REGISTRER: 注册类API</li> <li>• API_SPECIFIC_TYPE_ORCHESTRATE: 编排类API</li> </ul>
publish_status	否	String	API发布状态。 枚举值: <ul style="list-style-type: none"> <li>• PUBLISHED: 已发布</li> <li>• NOT_PUBLISHED: 未发布</li> </ul>
table_name	否	String	根据API用到的数据库表名模糊查询。

参数	是否必选	参数类型	描述
authorization_status	否	String	授权状态。 枚举值： <ul style="list-style-type: none"> <li>NO_AUTHORIZATION_REQUIRED: 无需授权</li> <li>UNAUTHORIZED: 未授权</li> <li>AUTHORIZED: 已授权</li> </ul>

## 请求参数

表 11-17 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。
x-return-publish-messages	否	String	是否返回专享版API的发布信息。

## 响应参数

状态码： 200

表 11-18 响应 Body 参数

参数	参数类型	描述
total	Integer	API总数量。

参数	参数类型	描述
records	Array of <a href="#">ApiOverview</a> objects	API列表。

表 11-19 ApiOverview

参数	参数类型	描述
id	String	API ID。
name	String	API名称。
group_id	String	API分组ID（共享版）。
description	String	API描述。
status	String	API的状态（共享版）。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_CREATED: API创建状态</li> <li>• API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>• API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>• API_STATUS_PUBLISHED: 已发布状态</li> <li>• API_STATUS_WAITING_STOP: 停用审核状态</li> <li>• API_STATUS_STOPPED: 已停用状态</li> <li>• API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>• API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>• API_STATUS_OFFLINE: 已下线</li> <li>• API_STATUS_OFFLINE_WAIT_REVIEW: 下线待审核状态</li> </ul>
debug_status	String	API调试状态（共享版）。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING: 等待调试</li> <li>• API_DEBUG_FAILED: 调试失败</li> <li>• API_DEBUG_SUCCESS: 调试成功</li> </ul>
publish_messages	Array of <a href="#">ApiPublishDTO</a> objects	发布信息列表（专享版）。

参数	参数类型	描述
type	String	API类型。 枚举值： <ul style="list-style-type: none"> <li>• API_SPECIFIC_TYPE_CONFIGURATION: 配置类API</li> <li>• API_SPECIFIC_TYPE_SCRIPT: 脚本类API</li> <li>• API_SPECIFIC_TYPE_REGISTER: 注册类API</li> </ul>
manager	String	API审核人。
create_user	String	API创建者。
create_time	Long	API创建时间。
authorization_status	String	授权状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_AUTHORIZATION_REQUIRED: 无需授权</li> <li>• UNAUTHORIZED: 未授权</li> <li>• AUTHORIZED: 已授权</li> </ul>

表 11-20 ApiPublishDTO

参数	参数类型	描述
id	String	发布编号。
api_id	String	API编号。
instance_id	String	集群ID编号。
instance_name	String	集群名称。

参数	参数类型	描述
api_status	String	API的状态。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_CREATED: API创建状态</li> <li>• API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>• API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>• API_STATUS_PUBLISHED: 已发布状态</li> <li>• API_STATUS_WAITING_STOP: 停用审核状态</li> <li>• API_STATUS_STOPPED: 已停用状态</li> <li>• API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>• API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>• API_STATUS_OFFLINE: 已下线</li> </ul>
api_debug	String	API调试状态。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING: 等待调试</li> <li>• API_DEBUG_FAILED: 调试失败</li> <li>• API_DEBUG_SUCCESS: 调试成功</li> </ul>

状态码： 400

表 11-21 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询ADMIN创建的已发布的脚本类API信息。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis?
api_type=API_SPECIFIC_TYPE_SCRIPT&publish_status=PUBLISHED&create_user=admin&description=测试
{
  "api_type": "API_SPECIFIC_TYPE_SCRIPT",
  "publish_status": "PUBLISHED",
  "create_user": "admin",
  "description": "测试"
}
```

## 响应示例

**状态码： 200**

Success。

```
{
  "total": 1,
  "records": [ {
    "id": "fd0beac79418d65e3b3ed24a6e53b24b",
    "name": "testApi",
    "create_time": 1579162215000,
    "status": "API_STATUS_CREATED",
    "manager": "admin",
    "type": "API_SPECIFIC_TYPE_REGISTER",
    "debug_status": "API_DEBUG_WAITING",
    "description": "api description"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询ADMIN创建的已发布的脚本类API信息。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListApisSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListApisRequest request = new ListApisRequest();
        try {
            ListApisResponse response = client.listApis(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

查询ADMIN创建的已发布的脚本类API信息。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListApisRequest()
        response = client.list_apis(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

查询ADMIN创建的已发布的脚本类API信息。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```

variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListApisRequest{}
response, err := client.ListApis(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

### 11.1.3 更新 API

#### 功能介绍

更新API。

#### 调用方法

请参见[如何调用API](#)。

#### URI

PUT /v1/{project\_id}/service/apis/{api\_id}



表 11-22 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API ID。

## 请求参数

表 11-23 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token, 使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型, 指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型(格式), 有Body体的情况下必选, 没有Body体无需填写。如果请求消息体中含有中文字符, 则需要通过charset=utf8指定中文字符集, 例如取值为: application/json;charset=utf8。

表 11-24 请求 Body 参数

参数	是否必选	参数类型	描述
catalog_id	否	String	目录ID。
name	否	String	API名称。
description	否	String	API描述。
log_flag	否	Boolean	是否启用访问日志。

参数	是否必选	参数类型	描述
api_type	否	String	API类型。 枚举值： <ul style="list-style-type: none"> <li>API_SPECIFIC_TYPE_CONFIGURATION: 配置类API</li> <li>API_SPECIFIC_TYPE_SCRIPT: 脚本类API</li> <li>API_SPECIFIC_TYPE_MYBATIS: MyBatis类API</li> <li>API_SPECIFIC_TYPE_GROOVY: Groovy类API</li> <li>API_SPECIFIC_TYPE_REGISTRER: 注册类API</li> <li>API_SPECIFIC_TYPE_ORCHESTRATE: 编排类API</li> <li>API_TYPE_CREATE: 创建类API (已弃用)</li> <li>API_TYPE_REGISTER: 注册类API (已弃用)</li> </ul>
auth_type	否	String	认证类型。 枚举值： <ul style="list-style-type: none"> <li>APP: app secret认证</li> <li>IAM: IAM token认证</li> <li>NONE: 无认证</li> </ul>
retention_period	否	Integer	最低保留期限，单位小时。
manager	否	String	API审核人。
path	否	String	API路径。
protocol	否	String	API请求协议类型。 枚举值： <ul style="list-style-type: none"> <li>PROTOCOL_TYPE_HTTP: HTTP协议请求</li> <li>PROTOCOL_TYPE_HTTPS: HTTPS协议请求</li> </ul>
request_type	否	String	请求类型。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_TYPE_POST: POST请求</li> <li>REQUEST_TYPE_GET: GET请求</li> </ul>

参数	是否必选	参数类型	描述
tags	否	Array of strings	标签。
visibility	否	String	API可见性。 枚举值： <ul style="list-style-type: none"> <li>• WORKSPACE: 工作空间可见</li> <li>• PROJECT: 项目可见</li> <li>• DOMAIN: 租户可见</li> </ul>
publish_type	否	String	发布类型(已弃用)。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH_TYPE_PUBLIC: 公开</li> <li>• PUBLISH_TYPE_PRIVATE: 私有</li> </ul>
data_mask_switch	否	Boolean	数据脱敏开关。
data_mask_params	否	Array of <b>DataMaskParams</b> objects	数据脱敏配置。
body	否	String	请求体内容描述，仅注册类API使用。
datasource_config_list	否	Array of <b>DatasourceConfig</b> objects	多数据源配置信息，仅Groovy类API使用。
groovy_content	否	String	Groovy文本，仅Groovy类API使用。
request_params	否	Array of <b>RequestParam</b> objects	API请求参数列表。
datasource_config	否	<b>DatasourceConfig</b> object	API数据源配置。
backend_config	否	<b>BackendConfig</b> object	API后端配置。仅注册类API使用

表 11-25 DataMaskParams

参数	是否必选	参数类型	描述
column_name	否	String	敏感字段。

参数	是否必选	参数类型	描述
algorithm_name	否	String	算法名称。
algorithm_type	否	String	算法类型。
en_name	否	String	算法名称。
algorithm_parameters	否	String	参数。
failure_policy	否	String	失败策略。 枚举值： <ul style="list-style-type: none"> <li>• SKIP: 跳过, 即不做处理</li> <li>• INTERRUPT_AND_EXCEPTION: 中断并抛出异常</li> <li>• SET_NULL: 置空</li> <li>• DEFAULT_VALUE: 默认值</li> </ul>

表 11-26 DatasourceConfig

参数	是否必选	参数类型	描述
type	否	String	数据源的类型。 枚举值： <ul style="list-style-type: none"> <li>• MYSQL: MYSQL数据源</li> <li>• DLI: DLI数据源</li> <li>• DWS: DWS数据源</li> <li>• HIVE: HIVE数据源</li> <li>• HBASE: HBASE数据源</li> </ul>
connection_name	否	String	数据连接名称。
connection_id	否	String	数据连接ID。
database	否	String	数据库名。
datatable	否	String	数据表名称。
table_id	否	String	数据表ID。
queue	否	String	DLI的队列名称。

参数	是否必选	参数类型	描述
access_mode	否	String	获取数据的模式。 枚举值： <ul style="list-style-type: none"> <li>• SQL: SQL取数方式</li> <li>• ROW_KEY: ROW_KEY取数方式 仅适用于HBASE</li> <li>• PREFIX_FILTER: PREFIX_FILTER取数方式仅适用于HBASE</li> </ul>
access_type	否	String	取数方式(已弃用)。 枚举值： <ul style="list-style-type: none"> <li>• SCRIPT: 脚本方式</li> <li>• CONFIGURAITON: 配置方式</li> </ul>
pagination	否	String	分页方式。 枚举值： <ul style="list-style-type: none"> <li>• DEFAULT: 默认分页</li> <li>• CUSTOM: 自定义分页</li> </ul>
sql	否	String	脚本模式下的SQL语句。
backend_params	否	Array of <a href="#">ApiRequestPara</a> objects	API后端参数。
response_params	否	Array of <a href="#">ApiResponsePara</a> objects	配置类API返回参数。
order_paras	否	Array of <a href="#">DatasourceOrderPara</a> objects	排序参数。
total_size_sql	否	String	总条数计算脚本SQL。

表 11-27 RequestPara

参数	是否必选	参数类型	描述
name	否	String	参数名。

参数	是否必选	参数类型	描述
position	否	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_PARAMETER_POSITION_PATH: 请求PATH内参数</li> <li>REQUEST_PARAMETER_POSITION_HEADER: 请求HEADER内参数</li> <li>REQUEST_PARAMETER_POSITION_QUERY: 请求QUERY参数</li> </ul>
type	否	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_PARAMETER_TYPE_NUMBER: 数值类型</li> <li>REQUEST_PARAMETER_TYPE_STRING: 字符串类型</li> </ul>
description	否	String	参数的描述。
necessary	否	Boolean	参数是否必填。
example_value	否	String	示例值。
default_value	否	String	默认值。
support_null	否	Boolean	支持NULL值。

表 11-28 ApiRequestPara

参数	是否必选	参数类型	描述
name	否	String	参数名称。
mapping	否	String	映射字段。

参数	是否必选	参数类型	描述
condition	否	String	操作符。 枚举值： <ul style="list-style-type: none"> <li>• CONDITION_TYPE_EQ: 等于</li> <li>• CONDITION_TYPE_NE: 不等于</li> <li>• CONDITION_TYPE_GT: 大于</li> <li>• CONDITION_TYPE_GE: 大于等于</li> <li>• CONDITION_TYPE_LT: 小于</li> <li>• CONDITION_TYPE_LE: 小于等于</li> <li>• CONDITION_TYPE_LIKE: 模糊查询如:%like%</li> <li>• CONDITION_TYPE_LIKE_L: 模糊查询如:%like</li> <li>• CONDITION_TYPE_LIKE_R: 模糊查询如:like%</li> </ul>

表 11-29 ApiResponsePara

参数	是否必选	参数类型	描述
name	否	String	参数名。
field	否	String	绑定的表字段。
type	否	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_TYPE_NUMBER: 数值类型</li> <li>• REQUEST_PARAMETER_TYPE_STRING: 字符串类型</li> </ul>
description	否	String	参数描述。
example_value	否	String	参数示例值。

表 11-30 DatasourceOrderPara

参数	是否必选	参数类型	描述
name	否	String	排序参数名称。
field	否	String	对应的参数字段。
optional	否	Boolean	是否可选。
sort	否	String	排序方式。 枚举值： <ul style="list-style-type: none"> <li>• ASC: 升序</li> <li>• DESC: 降序</li> <li>• CUSTOM: 自定义</li> </ul>
order	否	Integer	排序参数顺序。
description	否	String	排序参数描述信息。

表 11-31 BackendConfig

参数	是否必选	参数类型	描述
type	否	String	请求类型。 <ul style="list-style-type: none"> <li>• REQUEST_TYPE_POST: POST请求</li> <li>• REQUEST_TYPE_GET: GET请求</li> </ul>
protocol	否	String	API请求协议类型。 <ul style="list-style-type: none"> <li>• PROTOCOL_TYPE_HTTP: HTTP协议请求</li> <li>• PROTOCOL_TYPE_HTTPS: HTTPS协议请求</li> </ul>
host	否	String	后端HOST。
timeout	否	Integer	后端超时时间。
path	否	String	后端请求PATH。
backend_params	否	Array of BackendRequestPara objects	API后端参数。
constant_params	否	Array of BackendConstant objects	后端常量参数。



表 11-32 BackendRequestPara

参数	是否必选	参数类型	描述
name	否	String	API请求参数名称。
position	否	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_POSITION_PATH: 请求PATH内参数</li> <li>• REQUEST_PARAMETER_POSITION_HEADER: 请求HEADER内参数</li> <li>• REQUEST_PARAMETER_POSITION_QUERY: 请求QUERY参数</li> </ul>
backend_para_name	否	String	对应的后端参数。

表 11-33 BackendConstant

参数	是否必选	参数类型	描述
name	否	String	常量参数名。
type	否	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_TYPE_NUMBER: 数值类型</li> <li>• REQUEST_PARAMETER_TYPE_STRING: 字符串类型</li> </ul>
position	否	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_POSITION_PATH: 请求PATH内参数</li> <li>• REQUEST_PARAMETER_POSITION_HEADER: 请求HEADER内参数</li> <li>• REQUEST_PARAMETER_POSITION_QUERY: 请求QUERY参数</li> </ul>
description	否	String	常量参数描述。
value	否	String	常量参数值。

## 响应参数

状态码： 400

表 11-34 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

更新Id为10e3a4b67f59a9efe64e6bb31859221f的API信息。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/10e3a4b67f59a9efe64e6bb31859221f
{
  "catalog_id": "0",
  "name": "creat_test0010001",
  "description": "dws",
  "log_flag": false,
  "auth_type": "APP",
  "retention_period": 0,
  "publish_type": "PUBLISH_TYPE_PRIVATE",
  "manager": "user",
  "path": "/creat_test001/{p}",
  "api_type": "API_SPECIFIC_TYPE_SCRIPT",
  "protocol": "PROTOCOL_TYPE_HTTPS",
  "request_type": "REQUEST_TYPE_GET",
  "request_paras": [ {
    "name": "p",
    "position": "REQUEST_PARAMETER_POSITION_PATH",
    "type": "REQUEST_PARAMETER_TYPE_NUMBER",
    "description": "test",
    "necessary": true,
    "example_value": null,
    "support_null": true,
    "default_value": null
  } ],
  "datasource_config": {
    "type": "CLICKHOUSE",
    "connection_name": "clickhouse_ywx1119238_safe",
    "connection_id": "3d0d5941f2cb4ef08abecc01e33eabec",
    "database": "system",
    "queue": "dayu",
    "access_type": "CONNECTION_TYPE_SCRIPT",
    "access_mode": "SQL",
    "sql": "c2VsZWN0IDEGYXMgbnVtOw==",
    "backend_paras": [ ],
    "order_paras": [ ],
    "pagination": "DEFAULT"
  },
  "visibility": "WORKSPACE"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

更新Id为10e3a4b67f59a9efe64e6bb31859221f的API信息。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateApiSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateApiRequest request = new UpdateApiRequest();
        request.withApiId("{api_id}");
        Api body = new Api();
        DatasourceConfig datasourceConfigbody = new DatasourceConfig();
        datasourceConfigbody.withType(DatasourceConfig.TypeEnum.fromValue("CLICKHOUSE"))
            .withConnectionName("clickhouse_ywx1119238_safe")
            .withConnectionId("3d0d5941f2cb4ef08abec01e33eabec")
            .withDatabase("system")
            .withQueue("dayu")
            .withAccessType(DatasourceConfig.AccessTypeEnum.fromValue("CONNECTION_TYPE_SCRIPT"))
            .withAccessMode(DatasourceConfig.AccessModeEnum.fromValue("SQL"))
            .withPagination(DatasourceConfig.PaginationEnum.fromValue("DEFAULT"))
            .withSql("c2VsZWN0IDFgYXN0MgVtOw==");
        List<RequestPara> listbodyRequestParas = new ArrayList<>();
        listbodyRequestParas.add(
            new RequestPara()
                .withName("p")
                .withPosition(RequestPara.PositionEnum.fromValue("REQUEST_PARAMETER_POSITION_PATH"))
                .withType(RequestPara.TypeEnum.fromValue("REQUEST_PARAMETER_TYPE_NUMBER"))
                .withDescription("test")
                .withNecessary(true)
        );
        body.withDatasourceConfig(datasourceConfigbody);
        body.withRequestParas(listbodyRequestParas);
        body.withVisibility(Api.VisibilityEnum.fromValue("WORKSPACE"));
```

```
body.withRequestType(Api.RequestTypeEnum.fromValue("REQUEST_TYPE_GET"));
body.withProtocol(Api.ProtocolEnum.fromValue("PROTOCOL_TYPE_HTTPS"));
body.withPath("/creat_test001/{p}");
body.withManager("user");
body.withPublishType(Api.PublishTypeEnum.fromValue("PUBLISH_TYPE_PRIVATE"));
body.withAuthType(Api.AuthTypeEnum.fromValue("APP"));
body.withApiType(Api.ApiTypeEnum.fromValue("API_SPECIFIC_TYPE_SCRIPT"));
body.withLogFlag(false);
body.withDescription("dws");
body.withName("creat_test0010001");
body.withCatalogId("0");
request.withBody(body);
try {
    UpdateApiResponse response = client.updateApi(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

更新Id为10e3a4b67f59a9efe64e6bb31859221f的API信息。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateApiRequest()
        request.api_id = "{api_id}"
        datasourceConfigbody = DatasourceConfig(
            type="CLICKHOUSE",
            connection_name="clickhouse_ywx1119238_safe",
            connection_id="3d0d5941f2cb4ef08abec01e33eabec",
            database="system",
            queue="dayu",
            access_type="CONNECTION_TYPE_SCRIPT",
            access_mode="SQL",
```

```
        pagination="DEFAULT",
        sql="c2VsZWN0IDegYXMgbnVtOw=="
    )
    listRequestParasbody = [
        RequestPara(
            name="p",
            position="REQUEST_PARAMETER_POSITION_PATH",
            type="REQUEST_PARAMETER_TYPE_NUMBER",
            description="test",
            necessary=True
        )
    ]
    request.body = Api(
        datasource_config=datasourceConfigbody,
        request_paras=listRequestParasbody,
        visibility="WORKSPACE",
        request_type="REQUEST_TYPE_GET",
        protocol="PROTOCOL_TYPE_HTTPS",
        path="/creat_test001/{p}",
        manager="user",
        publish_type="PUBLISH_TYPE_PRIVATE",
        auth_type="APP",
        api_type="API_SPECIFIC_TYPE_SCRIPT",
        log_flag=False,
        description="dws",
        name="creat_test0010001",
        catalog_id="0"
    )
    response = client.update_api(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

更新Id为10e3a4b67f59a9efe64e6bb31859221f的API信息。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```

        WithCredential(auth).
        Build()

request := &model.UpdateApiRequest{}
request.ApiId = "{api_id}"
typeDatasourceConfig:= model.GetDatasourceConfigTypeEnum().CLICKHOUSE
connectionNameDatasourceConfig:= "clickhouse_ywx1119238_safe"
connectionIdDatasourceConfig:= "3d0d5941f2cb4ef08abec01e33eabec"
databaseDatasourceConfig:= "system"
queueDatasourceConfig:= "dayu"
accessTypeDatasourceConfig:=
model.GetDatasourceConfigAccessTypeEnum().CONNECTION_TYPE_SCRIPT
accessModeDatasourceConfig:= model.GetDatasourceConfigAccessModeEnum().SQL
paginationDatasourceConfig:= model.GetDatasourceConfigPaginationEnum().DEFAULT
sqlDatasourceConfig:= "c2VsZWN0IDEgYXMgbnVtOw=="
datasourceConfigbody := &model.DatasourceConfig{
    Type: &typeDatasourceConfig,
    ConnectionName: &connectionNameDatasourceConfig,
    ConnectionId: &connectionIdDatasourceConfig,
    Database: &databaseDatasourceConfig,
    Queue: &queueDatasourceConfig,
    AccessType: &accessTypeDatasourceConfig,
    AccessMode: &accessModeDatasourceConfig,
    Pagination: &paginationDatasourceConfig,
    Sql: &sqlDatasourceConfig,
}
nameRequestParas:= "p"
positionRequestParas:= model.GetRequestParaPositionEnum().REQUEST_PARAMETER_POSITION_PATH
typeRequestParas:= model.GetRequestParaTypeEnum().REQUEST_PARAMETER_TYPE_NUMBER
descriptionRequestParas:= "test"
necessaryRequestParas:= true
var listRequestParasbody = []model.RequestPara{
    {
        Name: &nameRequestParas,
        Position: &positionRequestParas,
        Type: &typeRequestParas,
        Description: &descriptionRequestParas,
        Necessary: &necessaryRequestParas,
    },
}
visibilityApi:= model.GetApiVisibilityEnum().WORKSPACE
requestTypeApi:= model.GetApiRequestTypeEnum().REQUEST_TYPE_GET
protocolApi:= model.GetApiProtocolEnum().PROTOCOL_TYPE_HTTPS
pathApi:= "/creat_test001/{p}"
managerApi:= "user"
publishTypeApi:= model.GetApiPublishTypeEnum().PUBLISH_TYPE_PRIVATE
authTypeApi:= model.GetApiAuthTypeEnum().APP
apiTypeApi:= model.GetApiApiTypeEnum().API_SPECIFIC_TYPE_SCRIPT
logFlagApi:= false
descriptionApi:= "dws"
nameApi:= "creat_test0010001"
catalogIdApi:= "0"
request.Body = &model.Api{
    DatasourceConfig: datasourceConfigbody,
    RequestParas: &listRequestParasbody,
    Visibility: &visibilityApi,
    RequestType: &requestTypeApi,
    Protocol: &protocolApi,
    Path: &pathApi,
    Manager: &managerApi,
    PublishType: &publishTypeApi,
    AuthType: &authTypeApi,
    ApiType: &apiTypeApi,
    LogFlag: &logFlagApi,
    Description: &descriptionApi,
    Name: &nameApi,
    CatalogId: &catalogIdApi,
}
response, err := client.UpdateApi(request)
    
```

```
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	Success。
400	BadRequest。

## 11.1.4 查询 API 信息

### 功能介绍

查询API信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/apis/{api\_id}

表 11-35 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API ID。

## 请求参数

表 11-36 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-37 响应 Body 参数

参数	参数类型	描述
id	String	API的ID。
name	String	API名称。
group_id	String	API所属分组的ID（共享版）。
description	String	API描述。
protocol	String	API请求协议类型。 枚举值： <ul style="list-style-type: none"> <li>PROTOCOL_TYPE_HTTP：HTTP协议请求</li> <li>PROTOCOL_TYPE_HTTPS：HTTPS协议请求</li> </ul>
publish_type	String	发布类型(已弃用)。 枚举值： <ul style="list-style-type: none"> <li>PUBLISH_TYPE_PUBLIC：公开</li> <li>PUBLISH_TYPE_PRIVATE：私有</li> </ul>



参数	参数类型	描述
log_flag	Boolean	是否开启日志记录。
path	String	API的访问路径。
host	String	共享版域名。
hosts	<b>InstanceHost DTO</b> object	专享版域名。
request_type	String	请求类型。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_TYPE_POST: POST请求</li> <li>REQUEST_TYPE_GET: GET请求</li> </ul>
create_user	String	API创建者。
create_time	Long	创建时间。
update_time	Long	更新时间。
manager	String	API审核人名称。
status	String	API的状态（共享版）。 枚举值： <ul style="list-style-type: none"> <li>API_STATUS_CREATED: API创建状态</li> <li>API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>API_STATUS_PUBLISHED: 已发布状态</li> <li>API_STATUS_WAITING_STOP: 停用审核状态</li> <li>API_STATUS_STOPPED: 已停用状态</li> <li>API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>API_STATUS_OFFLINE: 已下线</li> <li>API_STATUS_OFFLINE_WAIT_REVIEW: 下线待审核状态</li> </ul>
type	String	API类型。 枚举值： <ul style="list-style-type: none"> <li>API_SPECIFIC_TYPE_CONFIGURATION: 配置类API</li> <li>API_SPECIFIC_TYPE_SCRIPT: 脚本类API</li> <li>API_SPECIFIC_TYPE_REGISTER: 注册类API</li> </ul>

参数	参数类型	描述
debug_status	String	API调试状态（共享版）。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING：等待调试</li> <li>• API_DEBUG_FAILED：调试失败</li> <li>• API_DEBUG_SUCCESS：调试成功</li> </ul>
publish_messages	Array of <a href="#">ApiPublishDTO</a> objects	发布信息列表（专享版）。
request_params	Array of <a href="#">RequestParam</a> objects	API请求参数。
datasource_config	<a href="#">DataSourceConfig</a> object	数据源配置。
backend_config	<a href="#">BackendConfig</a> object	后端配置。

表 11-38 InstanceHostDTO

参数	参数类型	描述
instance_id	String	集群ID编号。
instance_name	String	集群名。
intranet_host	String	内网地址。
external_host	String	外网地址。
domains	Array of strings	网关域名。

表 11-39 ApiPublishDTO

参数	参数类型	描述
id	String	发布编号。
api_id	String	API编号。
instance_id	String	集群ID编号。
instance_name	String	集群名称。

参数	参数类型	描述
api_status	String	API的状态。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_CREATED: API创建状态</li> <li>• API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>• API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>• API_STATUS_PUBLISHED: 已发布状态</li> <li>• API_STATUS_WAITING_STOP: 停用审核状态</li> <li>• API_STATUS_STOPPED: 已停用状态</li> <li>• API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>• API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>• API_STATUS_OFFLINE: 已下线</li> </ul>
api_debug	String	API调试状态。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING: 等待调试</li> <li>• API_DEBUG_FAILED: 调试失败</li> <li>• API_DEBUG_SUCCESS: 调试成功</li> </ul>

表 11-40 RequestPara

参数	参数类型	描述
name	String	参数名。
position	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_POSITION_PATH: 请求PATH内参数</li> <li>• REQUEST_PARAMETER_POSITION_HEADER: 请求HEADER内参数</li> <li>• REQUEST_PARAMETER_POSITION_QUERY: 请求QUERY参数</li> </ul>
type	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_TYPE_NUMBER: 数值类型</li> <li>• REQUEST_PARAMETER_TYPE_STRING: 字符串类型</li> </ul>

参数	参数类型	描述
description	String	参数的描述。
necessary	Boolean	参数是否必填。
example_value	String	示例值。
default_value	String	默认值。
support_null	Boolean	支持NULL值。

表 11-41 DatasourceConfig

参数	参数类型	描述
type	String	数据源的类型。 枚举值： <ul style="list-style-type: none"> <li>• MYSQL: MYSQL数据源</li> <li>• DLI: DLI数据源</li> <li>• DWS: DWS数据源</li> <li>• HIVE: HIVE数据源</li> <li>• HBASE: HBASE数据源</li> </ul>
connection_name	String	数据连接名称。
connection_id	String	数据连接ID。
database	String	数据库名。
datatable	String	数据表名称。
table_id	String	数据表ID。
queue	String	DLI的队列名称。
access_mode	String	获取数据的模式。 枚举值： <ul style="list-style-type: none"> <li>• SQL: SQL取数方式</li> <li>• ROW_KEY: ROW_KEY取数方式 仅适用于HBASE</li> <li>• PREFIX_FILTER: PREFIX_FILTER取数方式仅适用于HBASE</li> </ul>
access_type	String	取数方式(已弃用)。 枚举值： <ul style="list-style-type: none"> <li>• SCRIPT: 脚本方式</li> <li>• CONFIGURAITON: 配置方式</li> </ul>

参数	参数类型	描述
pagination	String	分页方式。 枚举值： <ul style="list-style-type: none"> <li>• DEFAULT: 默认分页</li> <li>• CUSTOM: 自定义分页</li> </ul>
sql	String	脚本模式下的SQL语句。
backend_params	Array of <b>ApiRequestPara</b> objects	API后端参数。
response_params	Array of <b>ApiResponsePara</b> objects	配置类API返回参数。
order_params	Array of <b>DatasourceOrderPara</b> objects	排序参数。
total_size_sql	String	总条数计算脚本SQL。

表 11-42 ApiRequestPara

参数	参数类型	描述
name	String	参数名称。
mapping	String	映射字段。
condition	String	操作符。 枚举值： <ul style="list-style-type: none"> <li>• CONDITION_TYPE_EQ: 等于</li> <li>• CONDITION_TYPE_NE: 不等于</li> <li>• CONDITION_TYPE_GT: 大于</li> <li>• CONDITION_TYPE_GE: 大于等于</li> <li>• CONDITION_TYPE_LT: 小于</li> <li>• CONDITION_TYPE_LE: 小于等于</li> <li>• CONDITION_TYPE_LIKE: 模糊查询如:%like %</li> <li>• CONDITION_TYPE_LIKE_L: 模糊查询如:%like</li> <li>• CONDITION_TYPE_LIKE_R: 模糊查询如:like %</li> </ul>

表 11-43 ApiResponsePara

参数	参数类型	描述
name	String	参数名。
field	String	绑定的表字段。
type	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_PARAMETER_TYPE_NUMBER: 数值类型</li> <li>REQUEST_PARAMETER_TYPE_STRING: 字符串类型</li> </ul>
description	String	参数描述。
example_value	String	参数示例值。

表 11-44 DatasourceOrderPara

参数	参数类型	描述
name	String	排序参数名称。
field	String	对应的参数字段。
optional	Boolean	是否可选。
sort	String	排序方式。 枚举值： <ul style="list-style-type: none"> <li>ASC: 升序</li> <li>DESC: 降序</li> <li>CUSTOM: 自定义</li> </ul>
order	Integer	排序参数顺序。
description	String	排序参数描述信息。

表 11-45 BackendConfig

参数	参数类型	描述
type	String	请求类型。 <ul style="list-style-type: none"> <li>REQUEST_TYPE_POST: POST请求</li> <li>REQUEST_TYPE_GET: GET请求</li> </ul>

参数	参数类型	描述
protocol	String	API请求协议类型。 <ul style="list-style-type: none"> <li>• PROTOCOL_TYPE_HTTP: HTTP协议请求</li> <li>• PROTOCOL_TYPE_HTTPS: HTTPS协议请求</li> </ul>
host	String	后端HOST。
timeout	Integer	后端超时时间。
path	String	后端请求PATH。
backend_params	Array of <b>BackendRequestPara</b> objects	API后端参数。
constant_params	Array of <b>BackendConstant</b> objects	后端常量参数。

表 11-46 BackendRequestPara

参数	参数类型	描述
name	String	API请求参数名称。
position	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>• REQUEST_PARAMETER_POSITION_PATH: 请求PATH内参数</li> <li>• REQUEST_PARAMETER_POSITION_HEADER: 请求HEADER内参数</li> <li>• REQUEST_PARAMETER_POSITION_QUERY: 请求QUERY参数</li> </ul>
backend_param_name	String	对应的后端参数。

表 11-47 BackendConstant

参数	参数类型	描述
name	String	常量参数名。

参数	参数类型	描述
type	String	常量参数类型。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_PARAMETER_TYPE_NUMBER：数值类型</li> <li>REQUEST_PARAMETER_TYPE_STRING：字符串类型</li> </ul>
position	String	参数的位置。 枚举值： <ul style="list-style-type: none"> <li>REQUEST_PARAMETER_POSITION_PATH：请求PATH内参数</li> <li>REQUEST_PARAMETER_POSITION_HEADER：请求HEADER内参数</li> <li>REQUEST_PARAMETER_POSITION_QUERY：请求QUERY参数</li> </ul>
description	String	常量参数描述。
value	String	常量参数值。

状态码： 400

表 11-48 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询Id为10e3a4b67f59a9efe64e6bb31859221f的API信息。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/10e3a4b67f59a9efe64e6bb31859221f
```

## 响应示例

状态码： 200

Success。

```
{
  "id": "6b9e682fd6d0ef7c0f674701adb20762",
  "name": "testApi",
  "group_id": "677942eeca42b85d82fcef909d8345f4",
  "description": "",
  "protocol": "PROTOCOL_TYPE_HTTP",
  "publish_type": "PUBLISH_TYPE_PUBLIC",
```



```

"log_flag" : true,
"path" : "/testApi/{a}",
"host" : null,
"hosts" : null,
"request_type" : "REQUEST_TYPE_POST",
"create_time" : 1579162215000,
"update_time" : 1579162691000,
"manager" : "admin",
"status" : "API_STATUS_CREATED",
"type" : "API_SPECIFIC_TYPE_CONFIGURATION",
"debug_status" : "API_DEBUG_WAITING",
"request_paras" : [ {
  "name" : "a",
  "position" : "REQUEST_PARAMETER_POSITION_PATH",
  "type" : "REQUEST_PARAMETER_TYPE_NUMBER",
  "description" : "",
  "necessary" : false,
  "example_value" : "",
  "default_value" : ""
}, {
  "name" : "b",
  "position" : "REQUEST_PARAMETER_POSITION_HEADER",
  "type" : "REQUEST_PARAMETER_TYPE_NUMBER",
  "description" : "",
  "necessary" : false,
  "example_value" : "",
  "default_value" : ""
}, {
  "name" : "c",
  "position" : "REQUEST_PARAMETER_POSITION_QUERY",
  "type" : "REQUEST_PARAMETER_TYPE_NUMBER",
  "description" : "",
  "necessary" : false,
  "example_value" : "",
  "default_value" : ""
} ],
"datasource_config" : null,
"backend_config" : {
  "type" : "REQUEST_TYPE_GET",
  "protocol" : "PROTOCOL_TYPE_HTTP",
  "timeout" : "100",
  "host" : "www.baidu.com",
  "path" : "/testApi/{d}/{g}",
  "backend_paras" : [ {
    "name" : "a",
    "position" : "REQUEST_PARAMETER_POSITION_PATH",
    "backend_para_name" : "d"
  }, {
    "name" : "b",
    "position" : "REQUEST_PARAMETER_POSITION_QUERY",
    "backend_para_name" : "e"
  }, {
    "name" : "c",
    "position" : "REQUEST_PARAMETER_POSITION_HEADER",
    "backend_para_name" : "f"
  } ],
  "constant_paras" : [ {
    "name" : "g",
    "type" : "REQUEST_PARAMETER_TYPE_NUMBER",
    "position" : "REQUEST_PARAMETER_POSITION_PATH",
    "description" : "",
    "value" : "1"
  }, {
    "name" : "h",
    "type" : "REQUEST_PARAMETER_TYPE_NUMBER",
    "position" : "REQUEST_PARAMETER_POSITION_QUERY",
    "description" : "",
    "value" : "1"
  } ],
  {

```

```
"name" : "j",
"type" : "REQUEST_PARAMETER_TYPE_NUMBER",
"position" : "REQUEST_PARAMETER_POSITION_HEADER",
"description" : "",
"value" : "1"
}]
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowApiSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowApiRequest request = new ShowApiRequest();
        request.withApild("{api_id}");
        try {
            ShowApiResponse response = client.showApi(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowApiRequest()
        request.api_id = "{api_id}"
        response = client.show_api(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).  
Build())  
  
request := &model.ShowApiRequest{}  
request.ApiId = "{api_id}"  
response, err := client.ShowApi(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.1.5 批量删除 API

### 功能介绍

批量删除API。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/apis/batch-delete

表 11-49 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 11-50 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID</a> 和 <a href="#">工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-51 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of strings	删除API的ID列表。

## 响应参数

状态码： 400

表 11-52 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

批量删除API。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/batch-delete  
[ "62a02822ae8d06f38f4950bd680e2735" ]
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

批量删除API。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class DeleteApiSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        DeleteApiRequest request = new DeleteApiRequest();  
        List<String> listbodyApiDeleteDTO = new ArrayList<>();  
        listbodyApiDeleteDTO.add("62a02822ae8d06f38f4950bd680e2735");  
        request.withBody(listbodyApiDeleteDTO);  
        try {  
            DeleteApiResponse response = client.deleteApi(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
        }  
    }  
}
```

```
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

批量删除API。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteApiRequest()
        listApiDeleteDTObody = [
            "62a02822ae8d06f38f4950bd680e2735"
        ]
        request.body = listApiDeleteDTObody
        response = client.delete_api(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

批量删除API。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteApiRequest{}
var listApiDeleteDTObody = []string{
    "62a02822ae8d06f38f4950bd680e2735",
}
request.Body = &listApiDeleteDTObody
response, err := client.DeleteApi(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	Success。
400	BadRequest。

## 11.1.6 发布 API

### 功能介绍

发布API。API只有发布后，才能够被调用。API发布时，可以将API发送至指定网关。

- 共享版，必须发送至API网关共享版。
- 专享版，可以依据自身需要，选择将API发送至API网关专享版、ROMA-APIC、或不发布网关。

#### 说明

发布请求的发起者若非审核人，需要API的审核人完成申请的审核。



## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/service/apis/{api\_id}/instances/{instance\_id}/publish

表 11-53 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。
instance_id	是	String	集群ID编号。

## 请求参数

表 11-54 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-55 请求 Body 参数

参数	是否必选	参数类型	描述
apig_type	否	String	网关类型。 枚举值： <ul style="list-style-type: none"> <li>• APIG: APIG类型网关</li> <li>• APIGW: APIGW类型网关</li> <li>• ROMA_APIC: ROMA类型网关</li> </ul>
apig_instance_id	否	String	网关实例id。
group_id_in_apig	否	String	网关分组id。
roma_app_id	否	String	roma网关集成应用id。

## 响应参数

状态码： 400

表 11-56 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

发布ID为760715eb1bfce0c575abab3be3bd41e6的API到集群。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/760715eb1bfce0c575abab3be3bd41e6/instances/21398ikjsjd9087122d4e/publish
{
  "apig_type": "APIG",
  "apig_instance_id": "APIG",
  "group_id_in_apig": "c4ba07ad2ae14015921c36aa4136e14c"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

发布ID为760715eb1bfce0c575abab3be3bd41e6的API到集群。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class PublishApiToInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        PublishApiToInstanceRequest request = new PublishApiToInstanceRequest();
        request.withApiId("{api_id}");
        request.withInstanceId("{instance_id}");
        GatewayDTO body = new GatewayDTO();
        body.withGroupIdInApig("c4ba07ad2ae14015921c36aa4136e14c");
        body.withApigInstanceId("APIG");
        body.withApigType(GatewayDTO.ApigTypeEnum.fromValue("APIG"));
        request.withBody(body);
        try {
            PublishApiToInstanceResponse response = client.publishApiToInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

发布ID为760715eb1bfce0c575abab3be3bd41e6的API到集群。

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = PublishApiToInstanceRequest()
        request.api_id = "{api_id}"
        request.instance_id = "{instance_id}"
        request.body = GatewayDTO(
            group_id_in_apig="c4ba07ad2ae14015921c36aa4136e14c",
            apig_instance_id="APIG",
            apig_type="APIG"
        )
        response = client.publish_api_to_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

发布ID为760715eb1bfce0c575abab3be3bd41e6的API到集群。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.PublishApiToInstanceRequest{}
request.ApiId = "{api_id}"
request.InstanceId = "{instance_id}"
groupIdInApigGatewayDto:= "c4ba07ad2ae14015921c36aa4136e14c"
apigInstanceIdGatewayDto:= "APIG"
apigTypeGatewayDto:= model.GetGatewayDtoApigTypeEnum().APIG
request.Body = &model.GatewayDto{
    GroupIdInApig: &groupIdInApigGatewayDto,
    ApigInstanceId: &apigInstanceIdGatewayDto,
    ApigType: &apigTypeGatewayDto,
}
response, err := client.PublishApiToInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	操作API成功。
400	BadRequest。

## 11.1.7 API 操作(下线/停用/恢复)

### 功能介绍

- 下线API。将已发布的API下线。下线后，所有授权关系都会被解除，API将无法再被调用。
- 停用API。将已发布的API临时下线。下线后，授权关系会保留，停用期间API将无法再被调用。
- 恢复API。将已停用的API恢复使用。恢复后，API重新提供调用。

#### 说明

- 恢复请求的发起者若非审核人，需要API的审核人完成申请的审核。
- 下线/停用请求的发起者，必须为API的审核人。
- 下线/停用功能需要为已授权的应用预留充分的准备时间，需至少提前2天发起请求。若需要立即执行下线/停用，需要发起请求后，无有效的授权应用或是有效的授权应用均处理完消息（立即执行，或定期后完成执行）。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/service/apis/{api\_id}/instances/{instance\_id}/action

表 11-57 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。
instance_id	是	String	集群ID编号。

## 请求参数

表 11-58 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-59 请求 Body 参数

参数	是否必选	参数类型	描述
time	否	String	截止时间。

参数	是否必选	参数类型	描述
action	否	String	操作类型。 枚举值： <ul style="list-style-type: none"> <li>● UNPUBLISH：取消发布</li> <li>● STOP：停用</li> <li>● RECOVER：恢复</li> </ul>

## 响应参数

状态码： 400

表 11-60 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

下线/停用/恢复ID为51159105c7838353d2834181d978af50指定的集群上ID为760715eb1bfce0c575abab3be3bd41e6的API。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/760715eb1bfce0c575abab3be3bd41e6/instances/51159105c7838353d2834181d978af50/action
{
  "action" : "UNPUBLISH",
  "time" : "2022-03-25T10:00:00.000Z"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

下线/停用/恢复ID为51159105c7838353d2834181d978af50指定的集群上ID为760715eb1bfce0c575abab3be3bd41e6的API。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ExecuteApiToInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ExecuteApiToInstanceRequest request = new ExecuteApiToInstanceRequest();
        request.withApiId("{api_id}");
        request.withInstanceId("{instance_id}");
        ApiActionDTO body = new ApiActionDTO();
        body.withAction(ApiActionDTO.ActionEnum.fromValue("UNPUBLISH"));
        body.withTime("2022-03-25T10:00:00.000Z");
        request.withBody(body);
        try {
            ExecuteApiToInstanceResponse response = client.executeApiToInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

下线/停用/恢复ID为51159105c7838353d2834181d978af50指定的集群上ID为760715eb1bfce0c575abab3be3bd41e6的API。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```



```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ExecuteApiToInstanceRequest()
    request.api_id = "{api_id}"
    request.instance_id = "{instance_id}"
    request.body = ApiActionDTO(
        action="UNPUBLISH",
        time="2022-03-25T10:00:00.000Z"
    )
    response = client.execute_api_to_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

下线/停用/恢复ID为51159105c7838353d2834181d978af50指定的集群上ID为760715eb1bfce0c575abab3be3bd41e6的API。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExecuteApiToInstanceRequest{
        request.ApiId = "{api_id}"
        request.InstanceId = "{instance_id}"
        actionApiActionDto := model.GetApiActionDtoActionEnum().UNPUBLISH
    }
```

```
timeApiActionDto:= "2022-03-25T10:00:00.000Z"
request.Body = &model.ApiActionDto{
    Action: &actionApiActionDto,
    Time: &timeApiActionDto,
}
response, err := client.ExecuteApiToInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	操作API成功。
400	BadRequest。

## 11.1.8 批量授权 API(专享版)

### 功能介绍

APP创建成功后，还不能访问API，如果想要访问某个API，需要将该API授权给APP。API主动授权成功后，在有效期内，APP即可访问该API。API授权包含授权和续约两部分功能。

- 授权：授权会给予APP在有效期内访问API的权利。
- 续约：续约会更新授权有效期，仅支持延长有效期，不能减少。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/apis/{api\_id}/instances/{instance\_id}/authorize

表 11-61 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。

参数	是否必选	参数类型	描述
instance_id	是	String	集群ID编号。

## 请求参数

表 11-62 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必须选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-63 请求 Body 参数

参数	是否必选	参数类型	描述
time	否	String	截止时间。
app_ids	否	Array of strings	APP编号列表。

## 响应参数

状态码： 400

表 11-64 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

## 请求示例

授权/续约ID为51159105c7838353d2834181d978af50集群上ID为760715eb1bfce0c575abab3be3bd41e6的API。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/760715eb1bfce0c575abab3be3bd41e6/instances/51159105c7838353d2834181d978af50/authorize
{
  "app_ids": [ "efa895e32e758bd316726b196ca1e8de" ],
  "time": "2022-03-25T10:00:00.000Z"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

授权/续约ID为51159105c7838353d2834181d978af50集群上ID为760715eb1bfce0c575abab3be3bd41e6的API。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class AuthorizeApiToInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
AuthorizeApiToInstanceRequest request = new AuthorizeApiToInstanceRequest();
request.withApiId("{api_id}");
request.withInstanceId("{instance_id}");
ApiParaForAuthorizeToInstance body = new ApiParaForAuthorizeToInstance();
List<String> listbodyAppIds = new ArrayList<>();
listbodyAppIds.add("efa895e32e758bd316726b196ca1e8de");
body.withAppIds(listbodyAppIds);
body.withTime("2022-03-25T10:00:00.000Z");
request.withBody(body);
try {
    AuthorizeApiToInstanceResponse response = client.authorizeApiToInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

授权/续约ID为51159105c7838353d2834181d978af50集群上ID为760715eb1bfce0c575abab3be3bd41e6的API。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = AuthorizeApiToInstanceRequest()
        request.api_id = "{api_id}"
        request.instance_id = "{instance_id}"
        listAppIdsbody = [
            "efa895e32e758bd316726b196ca1e8de"
        ]
        request.body = ApiParaForAuthorizeToInstance(
            app_ids=listAppIdsbody,
```

```
        time="2022-03-25T10:00:00.000Z"
    )
    response = client.authorize_api_to_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

授权/续约ID为51159105c7838353d2834181d978af50集群上ID为760715eb1bfce0c575abab3be3bd41e6的API。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AuthorizeApiToInstanceRequest{}
    request.ApiId = "{api_id}"
    request.InstanceId = "{instance_id}"
    var listAppIdsbody = []string{
        "efa895e32e758bd316726b196ca1e8de",
    }
    timeApiParaForAuthorizeToInstance := "2022-03-25T10:00:00.000Z"
    request.Body = &model.ApiParaForAuthorizeToInstance{
        AppIds: &listAppIdsbody,
        Time: &timeApiParaForAuthorizeToInstance,
    }
    response, err := client.AuthorizeApiToInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	操作API成功。
400	BadRequest。

## 11.1.9 调试 API

### 功能介绍

调试API。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/apis/{api\_id}/instances/{instance\_id}/test

表 11-65 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。
instance_id	是	String	集群ID编号。

### 请求参数

表 11-66 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-67 请求 Body 参数

参数	是否必选	参数类型	描述
body	否	String	请求体。
paras	否	<a href="#">ApiTestParas</a> object	参数。

表 11-68 ApiTestParas

参数	是否必选	参数类型	描述
page_size	否	String	page size。
page_num	否	String	page num。

## 响应参数

状态码： 200

表 11-69 响应 Body 参数

参数	参数类型	描述
request_id	String	请求编号。
url	String	请求url。
result	String	调试结果。
timeout	Long	调试耗时。



参数	参数类型	描述
success	Boolean	是否调试成功。
request_header	ApiTestRequestHeader object	请求头。
response_header	ApiTestResponseHeader object	响应头。

表 11-70 ApiTestRequestHeader

参数	参数类型	描述
path	String	请求路径。
user_agent	String	代理（固定值）。
x_apig_mode	String	请求方式（固定值）。
x_app_identity	Integer	识别编号（固定值）。

表 11-71 ApiTestResponseHeader

参数	参数类型	描述
result_status	String	是否成功。
content_length	Integer	内容大小。
connection	String	连接状态。
cache_control	String	缓存控制（固定值）。
content_type	String	内容类型（固定值）。
date	String	日期。
x_request_id	String	请求ID。

状态码： 400

表 11-72 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

## 请求示例

调试API。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/760715eb1bfce0c575abab3be3bd41e6/instances/51159105c7838353d2834181d978af50/test
```

```
{  
  "body": "b",  
  "paras": {  
    "page_size": "10",  
    "page_num": "1"  
  }  
}
```

## 响应示例

**状态码: 200**

Success。

```
{  
  "request_header": {  
    "path": "GET /testApi/1 HTTP",  
    "user_agent": "DLMDebugClient/1.0",  
    "x_apig_mode": "DEBUG",  
    "x_app_identity": 1  
  },  
  "request_id": "704742365000242752",  
  "response_header": {  
    "cache_control": "private",  
    "connection": "keep-alive",  
    "content_length": 2,  
    "content_type": "application/json;charset=UTF-8",  
    "date": "Tue Apr 28 17:14:37 GMT+08:00 2020",  
    "result_status": "HTTP 200 Success",  
    "x_request_id": "704742365000242752"  
  },  
  "result": "[]",  
  "success": true,  
  "timeout": 3826,  
  "url": "/testApi/1"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

调试API。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DebugApiSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DebugApiRequest request = new DebugApiRequest();
        request.withApiId("{api_id}");
        request.withInstanceId("{instance_id}");
        ApiTestDTO body = new ApiTestDTO();
        ApiTestParas parasbody = new ApiTestParas();
        parasbody.withPageSize("10")
            .withPageNum("1");
        body.withParas(parasbody);
        body.withBody("b");
        request.withBody(body);
        try {
            DebugApiResponse response = client.debugApi(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

调试API。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DebugApiRequest()
    request.api_id = "{api_id}"
    request.instance_id = "{instance_id}"
    parasbody = ApiTestParas(
        page_size="10",
        page_num="1"
    )
    request.body = ApiTestDTO(
        paras=parasbody,
        body="b"
    )
    response = client.debug_api(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

调试API。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```

Build()

request := &model.DebugApiRequest{}
request.ApiId = "{api_id}"
request.InstanceId = "{instance_id}"
pageSizeParas:= "10"
pageNumParas:= "1"
parasbody := &model.ApiTestParas{
    PageSize: &pageSizeParas,
    PageNum: &pageNumParas,
}
bodyApiTestDto:= "b"
request.Body = &model.ApiTestDto{
    Paras: parasbody,
    Body: &bodyApiTestDto,
}
response, err := client.DebugApi(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.1.10 API 授权操作(授权/取消授权/申请/续约)

### 功能介绍

- API主动授权：API审核人可发起，API主动授权成功后，在有效期内，APP即可访问该API。API授权包含授权和续约两部分功能。
  - 授权：授权会给予APP在有效期内访问API的权利。
  - 续约：续约会更新授权有效期，仅支持延长有效期，不能减少。
- API解除授权：API审核人可发起，解除API对APP的授权关系。解除授权后，APP将不再能够调用该API。API解除已授权的APP关系，需要为APP预留至少2天的准备时间。
- APP解除授权：APP所有者可发起，解除API对APP的授权关系。解除授权后，APP将不再能够调用该API。APP解除自己的授权关系，无需预留准备时间。
- APP申请授权：APP所有者可发起，APP申请API后，待API的审核人完成审核，APP即可访问该API。授权会给予APP在有效期内访问API的权利，需要API审核。
- APP申请续约：APP所有者可发起，续约会更新授权有效期，仅支持延长有效期，不能减少，需要API审核。

### 📖 说明

- 申请自己的API推荐采用API主动授权/续约，无需审核。
- 自己的应用推荐采用APP解除授权，无需预留准备时间。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/service/apis/authorize/action

表 11-73 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 11-74 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-75 请求 Body 参数

参数	是否必选	参数类型	描述
api_id	否	String	API编号。
instance_id	否	String	集群ID编号。
app_id	否	String	APP编号。
apply_type	否	String	申请类型。 枚举值： <ul style="list-style-type: none"> <li>• APPLY_TYPE_AUTHORIZE: API主动授权</li> <li>• APPLY_TYPE_API_CANCEL_AUTHORIZE: API取消授权</li> <li>• APPLY_TYPE_APP_CANCEL_AUTHORIZE: APP取消授权</li> <li>• APPLY_TYPE_APPLY: 审核 API</li> <li>• APPLY_TYPE_RENEW: 续约 API</li> </ul>
time	否	String	截止时间。

## 响应参数

状态码： 400

表 11-76 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

API主动授权/API解除授权/APP解除授权/APP申请授权/APP申请续约。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/authorize/action
```

```
{
  "api_id": "47046fe7830c1be77cb0dc23bd86afa5",
  "instance_id": "21398ikjdsjd9087122d4e",
  "app_id": "908489209a320df61607355c57c82882",
  "apply_type": "APPLY_TYPE_AUTHORIZE",
  "time": "2021-01-01T10:00:00.000Z"
}
```

## 响应示例

无

## 状态码

状态码	描述
204	操作API成功。
400	BadRequest。

## 11.1.11 查看 API 发布信息(专享版)

### 功能介绍

查看API在不同集群上的发布信息。API在集群上进行过操作后会存在发布信息，例如调试、注册类发布等。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/apis/{api\_id}/publish-info

表 11-77 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。

表 11-78 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或LIMIT的整数倍, 不满足则向下取整。



## 请求参数

表 11-79 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-80 响应 Body 参数

参数	参数类型	描述
total	Integer	总数。
publish_messages	Array of <a href="#">ApiPublishDTO</a> objects	发布信息列表。

表 11-81 ApiPublishDTO

参数	参数类型	描述
id	String	发布编号。
api_id	String	API编号。
instance_id	String	集群ID编号。

参数	参数类型	描述
instance_name	String	集群名称。
api_status	String	API的状态。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_CREATED: API创建状态</li> <li>• API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>• API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>• API_STATUS_PUBLISHED: 已发布状态</li> <li>• API_STATUS_WAITING_STOP: 停用审核状态</li> <li>• API_STATUS_STOPPED: 已停用状态</li> <li>• API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>• API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>• API_STATUS_OFFLINE: 已下线</li> </ul>
api_debug	String	API调试状态。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING: 等待调试</li> <li>• API_DEBUG_FAILED: 调试失败</li> <li>• API_DEBUG_SUCCESS: 调试成功</li> </ul>

状态码： 400

表 11-82 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查看API在不同集群上的发布信息。

`/v1/0833a5737480d53b2f250010d01a7b88/service/apis/760715eb1bfce0c575abab3be3bd41e6/publish-info`

## 响应示例

状态码： 200

Success。

```
{
  "total": 1,
  "publish_messages": [ {
    "id": "343a6d4c0ad108376ebd060c9c33cc33",
    "api_id": "47046fe7830c1be77cb0dc23bd86afa5",
    "instance_id": "c2e27497562ed193583378e052248003",
    "instance_name": "cluster_test_001",
    "api_status": "API_STATUS_CREATED",
    "api_debug": "API_DEBUG_WAITING"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchPublishInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        SearchPublishInfoRequest request = new SearchPublishInfoRequest();
        request.withApiId("{api_id}");
        try {
            SearchPublishInfoResponse response = client.searchPublishInfo(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = SearchPublishInfoRequest()  
        request.api_id = "{api_id}"  
        response = client.search_publish_info(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.SearchPublishInfoRequest{}
request.ApiId = "{api_id}"
response, err := client.SearchPublishInfo(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.1.12 查看 API 不同操作对应的实例信息(专享版)

### 功能介绍

查看API不同操作对应的实例信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/apis/{api\_id}/instances

表 11-83 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。

表 11-84 Query 参数

参数	是否必选	参数类型	描述
action	是	String	API操作。 枚举值： <ul style="list-style-type: none"> <li>• PUBLISH: 发布API</li> <li>• UNPUBLISH: 取消发布API</li> <li>• STOP: 停用API</li> <li>• RECOVER: 恢复API</li> <li>• WHITELIST: 白名单操作</li> <li>• AUTHORIZE: 授权操作</li> </ul>
show_all	否	Boolean	全部展示(包括不可执行当前操作的实例)。
check_status	否	Boolean	校验api状态。
check_debug	否	Boolean	校验API调试状态。
app_id	否	String	APP编号(用于判断授权操作app可选的实例)。
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或LIMIT的整数倍, 不满足则向下取整。

## 请求参数

表 11-85 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token, 使用 <b>Token认证</b> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型, 指定SHARED共享版或EXCLUSIVE专享版。

参数	是否必选	参数类型	描述
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-86 响应 Body 参数

参数	参数类型	描述
instances	Array of <b>InstanceForApiActionDTO</b> objects	api操作对应的实例列表。

表 11-87 InstanceForApiActionDTO

参数	参数类型	描述
instance_id	String	集群ID编号。
instance_type	String	集群类型。 枚举值： <ul style="list-style-type: none"> <li>● DLM：数据服务集群</li> <li>● APIG：APIG类型集群</li> <li>● APIGW：APIGW类型集群</li> <li>● ROMA_APIC：ROMA类型集群</li> </ul>
name	String	集群名称。
action	String	API操作。 枚举值： <ul style="list-style-type: none"> <li>● PUBLISH：发布API</li> <li>● UNPUBLISH：取消发布API</li> <li>● STOP：停用API</li> <li>● RECOVER：恢复API</li> <li>● WHITELIST：白名单操作</li> <li>● AUTHORIZE：授权操作</li> </ul>

参数	参数类型	描述
result	Boolean	校验结果。
cause	String	校验失败的原因。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_ERROR: API状态错误</li> <li>• API_DEBUG_ERROR: API调试状态错误</li> <li>• TYPE_MISMATCH: APP与实例类型不匹配</li> </ul>
api_status	String	API状态。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_CREATED: API创建状态</li> <li>• API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>• API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>• API_STATUS_PUBLISHED: 已发布状态</li> <li>• API_STATUS_WAITING_STOP: 停用审核状态</li> <li>• API_STATUS_STOPPED: 已停用状态</li> <li>• API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>• API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>• API_STATUS_OFFLINE: 已下线</li> </ul>
api_debug	String	API调试状态。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING: 等待调试</li> <li>• API_DEBUG_FAILED: 调试失败</li> <li>• API_DEBUG_SUCCESS: 调试成功</li> </ul>

状态码： 400

表 11-88 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查看API不同操作对应的实例信息(专享版)。



```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/760715eb1bfce0c575abab3be3bd41e6/instances
```

## 响应示例

**状态码： 200**

Success。

```
{
  "instances" : [ {
    "instance_id" : "c2e27497562ed193583378e052248003",
    "instance_type" : "DLM",
    "name" : "cluster_test_001",
    "action" : "PUBLISH",
    "result" : false,
    "cause" : "API_STATUS_ERROR",
    "api_status" : "API_STATUS_CREATED",
    "api_debug" : "API_DEBUG_WAITING"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListInstanceListSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListInstanceListRequest request = new ListInstanceListRequest();
        request.withApiId("{api_id}");
        try {
            ListInstanceListResponse response = client.listInstanceList(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstanceListRequest()
        request.api_id = "{api_id}"
        response = client.list_instance_list(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListInstanceListRequest{
    request.ApiId = "{api_id}"
}
response, err := client.ListInstanceList(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

### 11.1.13 查看 API 调试信息(专享版)

#### 功能介绍

查看API在不同集群上的调试信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/service/apis/{api\_id}/debug-info

表 11-89 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。

表 11-90 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或LIMIT的整数倍, 不满足则向下取整。

## 请求参数

表 11-91 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-92 响应 Body 参数

参数	参数类型	描述
total	Integer	总数。
publish_messages	Array of <a href="#">ApiPublishDTO</a> objects	发布信息列表。

表 11-93 ApiPublishDTO

参数	参数类型	描述
id	String	发布编号。
api_id	String	API编号。
instance_id	String	集群ID编号。
instance_name	String	集群名称。
api_status	String	API的状态。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_CREATED: API创建状态</li> <li>• API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>• API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>• API_STATUS_PUBLISHED: 已发布状态</li> <li>• API_STATUS_WAITING_STOP: 停用审核状态</li> <li>• API_STATUS_STOPPED: 已停用状态</li> <li>• API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>• API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>• API_STATUS_OFFLINE: 已下线</li> </ul>
api_debug	String	API调试状态。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING: 等待调试</li> <li>• API_DEBUG_FAILED: 调试失败</li> <li>• API_DEBUG_SUCCESS: 调试成功</li> </ul>

状态码： 400

表 11-94 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查看API调试信息(专享版)。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apis/760715eb1bfce0c575abab3be3bd41e6/debug-info
```

## 响应示例

**状态码： 200**

Success。

```
{
  "total": 1,
  "publish_messages": [ {
    "id": "343a6d4c0ad108376ebd060c9c33cc33",
    "api_id": "47046fe7830c1be77cb0dc23bd86afa5",
    "instance_id": "c2e27497562ed193583378e052248003",
    "instance_name": "cluster_test_001",
    "api_status": "API_STATUS_CREATED",
    "api_debug": "API_DEBUG_WAITING"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchDebugInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
SearchDebugInfoRequest request = new SearchDebugInfoRequest();
request.withApild("{api_id}");
try {
    SearchDebugInfoResponse response = client.searchDebugInfo(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

## Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchDebugInfoRequest()
        request.api_id = "{api_id}"
        response = client.search_debug_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchDebugInfoRequest{}
    request.ApiId = "{api_id}"
    response, err := client.SearchDebugInfo(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.1.14 导入包含 API 信息的 EXCEL 文件

### 功能介绍

导入包含API信息的EXCEL文件。



## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/service/import/excel

表 11-95 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 11-96 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），此处为导入文件，取值为multipart/form-data。

表 11-97 FormData 参数

参数	是否必选	参数类型	描述
file	是	File	导入包含API的EXCEL文件，大小必须小于4MB。

## 响应参数

状态码： 200

表 11-98 响应 Body 参数

参数	参数类型	描述
job_id	String	返回的数据信息。

状态码： 400

表 11-99 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

导入包含API信息的EXCEL文件。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/import/excel
```

## 响应示例

状态码： 200

导入EXCEL成功。

```
{
  "job_id": "252041f9838e8e5ea2e0e7317d5bcaf3"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ImportDataServiceExcelSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
```

```
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ImportDataServiceExcelRequest request = new ImportDataServiceExcelRequest();
try {
    ImportDataServiceExcelResponse response = client.importDataServiceExcel(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ImportDataServiceExcelRequest()
        response = client.import_data_service_excel(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ImportDataServiceExcelRequest{}
    response, err := client.ImportDataServiceExcel(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	导入EXCEL成功。
400	BadRequest。

## 11.1.15 导出包含 API 信息的 EXCEL 文件

### 功能介绍

导出包含API信息的EXCEL文件。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/service/export/excel

表 11-100 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 11-101 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。

表 11-102 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of strings	导出EXCEL文件。

## 响应参数

状态码： 200

表 11-103 响应 Body 参数

参数	参数类型	描述
-	File	导出包含API的EXCEL文件，大小必须小于4MB。

状态码： 400

表 11-104 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

导出包含API信息的EXCEL文件。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/export/excel
["6b9e682fd6d0ef7c0f674701adb20762", "42505b429b53b42a9b535640617d9b17" ]
```

## 响应示例

状态码： 200

导出EXCEL成功。

```
"xxx.xlsx"
```

## SDK 代码示例

SDK代码示例如下。

### Java

导出包含API信息的EXCEL文件。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ExportDataServiceExcelSolution {

    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ExportDataServiceExcelRequest request = new ExportDataServiceExcelRequest();
List<String> listbodyApiExportDTO = new ArrayList<>();
listbodyApiExportDTO.add("6b9e682fd6d0ef7c0f674701adb20762");
listbodyApiExportDTO.add("42505b429b53b42a9b535640617d9b17");
request.withBody(listbodyApiExportDTO);
try {
    ExportDataServiceExcelResponse response = client.exportDataServiceExcel(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

导出包含API信息的EXCEL文件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = ExportDataServiceExcelRequest()
    listApiExportDTObody = [
        "6b9e682fd6d0ef7c0f674701adb20762",
        "42505b429b53b42a9b535640617d9b17"
    ]
    request.body = listApiExportDTObody
    response = client.export_data_service_excel(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

导出包含API信息的EXCEL文件。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExportDataServiceExcelRequest{}
    var listApiExportDTObody = []string{
        "6b9e682fd6d0ef7c0f674701adb20762",
        "42505b429b53b42a9b535640617d9b17",
    }
    request.Body = &listApiExportDTObody
    response, err := client.ExportDataServiceExcel(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```



## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	导出EXCEL成功。
400	BadRequest。

## 11.1.16 全量导出包含 API 的 EXCEL 压缩文件

### 功能介绍

全量导出包含API的EXCEL压缩文件。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/export/zip

表 11-105 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 11-106 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
Dlm-Type	否	String	数据服务的版本类型，指定 SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有 Body体的情况下必选，没有 Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-107 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	API名称，该参数为模糊匹配。
create_user	否	String	API创建人名称。
description	否	String	API描述。
tags	否	Array of strings	API标签列表。
table_name	否	String	API所使用到的数据库表名。
publish_status_type	否	String	API发布状态。 枚举值： <ul style="list-style-type: none"> <li>● PUBLISHED: 已发布</li> <li>● NOT_PUBLISHED: 未发布</li> </ul>
api_specific_type_str	否	String	API取数方式。 枚举值： <ul style="list-style-type: none"> <li>● API_SPECIFIC_TYPE_CONFIGURATION: 配置方式</li> <li>● API_SPECIFIC_TYPE_SCRIPT: 脚本方式</li> <li>● API_SPECIFIC_TYPE_MYBATIS: MyBatis方式</li> <li>● API_SPECIFIC_TYPE_GROOVY: Groovy方式</li> </ul>
start_time	否	String	API创建开始时间。
end_time	否	String	API创建结束时间。

参数	是否必选	参数类型	描述
authorization_status_type	否	String	授权状态。 枚举值： <ul style="list-style-type: none"> <li>NO_AUTHORIZATION_REQUIRED: 无需授权</li> <li>UNAUTHORIZED: 未授权</li> <li>AUTHORIZED: 已授权</li> </ul>

## 响应参数

状态码： 200

表 11-108 响应 Body 参数

参数	参数类型	描述
-	File	导出包含API的EXCEL文件，会导出多个EXCEL文件，每个EXCEL文件最多200个API，多个EXCEL文件会打包到ZIP文件返回。

状态码： 400

表 11-109 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

全量导出包含API的EXCEL压缩文件。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/export/zip
{
  "name": "name",
  "create_user": "dev",
  "api_specific_type_str": "API_SPECIFIC_TYPE_CONFIGURATION",
  "start_time": "2024-02-29T16:00:00.000Z",
  "end_time": "2024-07-02T15:59:59.998Z",
  "tags": [ "111", "22", "33" ]
}
```

## 响应示例

状态码： 200

全量导出成功。

```
"xxx.zip"
```

## SDK 代码示例

SDK代码示例如下。

### Java

全量导出包含API的EXCEL压缩文件。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ExportDataServiceZipSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ExportDataServiceZipRequest request = new ExportDataServiceZipRequest();
        ApiParam body = new ApiParam();
        List<String> listbodyTags = new ArrayList<>();
        listbodyTags.add("111");
        listbodyTags.add("22");
        listbodyTags.add("33");
        body.withEndTime("2024-07-02T15:59:59.998Z");
        body.withStartTime("2024-02-29T16:00:00.000Z");

        body.withApiSpecificTypeStr(ApiParam.ApiSpecificTypeEnum.fromValue("API_SPECIFIC_TYPE_CONFIGURA
TION"));
        body.withTags(listbodyTags);
        body.withCreateUser("dev");
        body.withName("name");
        request.withBody(body);
        try {
            ExportDataServiceZipResponse response = client.exportDataServiceZip(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

全量导出包含API的EXCEL压缩文件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ExportDataServiceZipRequest()
        listTagsbody = [
            "111",
            "22",
            "33"
        ]
        request.body = ApiParam(
            end_time="2024-07-02T15:59:59.998Z",
            start_time="2024-02-29T16:00:00.000Z",
            api_specific_type_str="API_SPECIFIC_TYPE_CONFIGURATION",
            tags=listTagsbody,
            create_user="dev",
            name="name"
        )
        response = client.export_data_service_zip(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

全量导出包含API的EXCEL压缩文件。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExportDataServiceZipRequest{
        var listTagsbody = []string{
            "111",
            "22",
            "33",
        }
        endTimeApiParam:= "2024-07-02T15:59:59.998Z"
        startTimeApiParam:= "2024-02-29T16:00:00.000Z"
        apiSpecificTypeStrApiParam:=
        model.GetApiParamApiSpecificTypeEnum().API_SPECIFIC_TYPE_CONFIGURATION
        createUserApiParam:= "dev"
        nameApiParam:= "name"
        request.Body = &model.ApiParam{
            EndTime: &endTimeApiParam,
            StartTime: &startTimeApiParam,
            ApiSpecificTypeStr: &apiSpecificTypeStrApiParam,
            Tags: &listTagsbody,
            CreateUser: &createUserApiParam,
            Name: &nameApiParam,
        }
    }
    response, err := client.ExportDataServiceZip(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	全量导出成功。
400	BadRequest。

## 11.1.17 下载 EXCEL 模板

### 功能介绍

下载EXCEL模板。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/export/excel-template

表 11-110 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 11-111 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。

## 响应参数

状态码： 200

表 11-112 响应 Body 参数

参数	参数类型	描述
-	File	下载EXCEL模板。

状态码： 400

表 11-113 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

下载EXCEL模板。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/export/excel-template
```

## 响应示例

状态码： 200

下载EXCEL模板成功。

```
"xxx.xlsx"
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ExportDataServiceExcelTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```



security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ExportDataServiceExcelTemplateRequest request = new ExportDataServiceExcelTemplateRequest();
try {
    ExportDataServiceExcelTemplateResponse response =
client.exportDataServiceExcelTemplate(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ExportDataServiceExcelTemplateRequest()
        response = client.export_data_service_excel_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExportDataServiceExcelTemplateRequest{}
    response, err := client.ExportDataServiceExcelTemplate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	下载EXCEL模板成功。
400	BadRequest。

## 11.2 申请管理接口

## 11.2.1 查询申请列表

### 功能介绍

查询申请列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/applys

表 11-114 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-115 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	查询起始坐标，即跳过前X条数据。仅支持0或limit的整数倍，不满足则向下取整。
limit	否	Integer	查询条数限制。
api_name	否	String	API名称，该参数为模糊匹配。
query_type	否	Integer	查询类型，0:收到的申请(待审核)，1:收到的申请(已审核)，2:发出的申请(开发)，3:发出的申请(调用)。

### 请求参数

表 11-116 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-117 响应 Body 参数

参数	参数类型	描述
total	Integer	符合条件的申请总数。
records	Array of <a href="#">RecordForApplyDetail</a> objects	本次返回的申请列表。

表 11-118 RecordForApplyDetail

参数	参数类型	描述
id	String	申请编号。

参数	参数类型	描述
api_apply_status	String	申请状态。 枚举值： <ul style="list-style-type: none"> <li>STATUS_TYPE_PENDING_APPROVAL: 待审核</li> <li>STATUS_TYPE_REJECTED: 已驳回</li> <li>STATUS_TYPE_PENDING_CHECK: 待确认</li> <li>STATUS_TYPE_PENDING_EXECUTE: 待执行</li> <li>STATUS_TYPE_SYNCHRONOUS_EXECUTE: 同步执行</li> <li>STATUS_TYPE_FORCED_CANCEL: 强制取消</li> <li>STATUS_TYPE_PASSED: 已通过</li> </ul>
api_apply_type	String	申请类型。 枚举值： <ul style="list-style-type: none"> <li>APPLY_TYPE_PUBLISH: 发布API</li> <li>APPLY_TYPE_AUTHORIZE: API主动授权</li> <li>APPLY_TYPE_APPLY: 审核API</li> <li>APPLY_TYPE_RENEW: 申请续约API</li> <li>APPLY_TYPE_STOP: 申请停用API</li> <li>APPLY_TYPE_RECOVER: 申请恢复API</li> <li>APPLY_TYPE_API_CANCEL_AUTHORIZE: API取消授权</li> <li>APPLY_TYPE_APP_CANCEL_AUTHORIZE: APP取消授权</li> <li>APPLY_TYPE_OFFLINE: 申请下线</li> </ul>
api_id	String	API编号。
api_name	String	API名称。
api_using_time	Long	使用截止时间。
app_id	String	APP编号。
app_name	String	APP名称。
apply_time	Long	申请时间。
approval_time	Long	授权时间。
approver_name	String	审核人名称。
comment	String	审核评论。
user_name	String	申请人姓名。

状态码： 400

表 11-119 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询申请列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/applies
```

## 响应示例

状态码： 200

查询结果成功，符合条件的申请总数。

```
{
  "total": 1,
  "records": [ {
    "id": "a1c93c3cb17d9747fc41df0377b6fe0f",
    "api_apply_status": "STATUS_TYPE_PASSED",
    "api_apply_type": "APPLY_TYPE_APPLY",
    "api_id": "47046fe7830c1be77cb0dc23bd86afa5",
    "api_name": "api_demo",
    "api_using_time": 1580452617000,
    "app_id": "908489209a320df61607355c57c82882",
    "app_name": "app_demo",
    "apply_time": 1578875421000,
    "approval_time": 1578897421000,
    "approver_name": "李四",
    "comment": "通过",
    "user_name": "张三"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class ListApplySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListApplyRequest request = new ListApplyRequest();
        try {
            ListApplyResponse response = client.listApply(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListApplyRequest()
```

```

response = client.list_apply(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListApplyRequest{}
    response, err := client.ListApply(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询结果成功，符合条件的申请总数。
400	BadRequest。



## 11.2.2 审核申请

### 功能介绍

审核申请。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/applys

表 11-120 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 11-121 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-122 请求 Body 参数

参数	是否必选	参数类型	描述
apply_result	否	Boolean	申请结果。
apply_ids	否	Array of strings	申请编号列表。

## 响应参数

状态码： 400

表 11-123 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

对发起的申请进行审核通过或驳回。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/applies
{
  "apply_result" : false,
  "apply_ids" : [ "05b90cb05eb6bd7fcfd834dfc290c3ce" ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

对发起的申请进行审核通过或驳回。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
```

```
import java.util.ArrayList;

public class BatchApproveApplySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchApproveApplyRequest request = new BatchApproveApplyRequest();
        OpenApplyIdsForApproveApply body = new OpenApplyIdsForApproveApply();
        List<String> listbodyApplyIds = new ArrayList<>();
        listbodyApplyIds.add("05b90cb05eb6bd7fcfd834dfc290c3ce");
        body.withApplyIds(listbodyApplyIds);
        body.withApplyResult(false);
        request.withBody(body);
        try {
            BatchApproveApplyResponse response = client.batchApproveApply(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

对发起的申请进行审核通过或驳回。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = BatchApproveApplyRequest()
    listApplyIdsbody = [
        "05b90cb05eb6bd7fcfd834dfc290c3ce"
    ]
    request.body = OpenApplyIdsForApproveApply(
        apply_ids=listApplyIdsbody,
        apply_result=False
    )
    response = client.batch_approve_apply(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

对发起的申请进行审核通过或驳回。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchApproveApplyRequest{}
    var listApplyIdsbody = []string{
        "05b90cb05eb6bd7fcfd834dfc290c3ce",
    }
    applyResultOpenApplyIdsForApproveApply := false
    request.Body = &model.OpenApplyIdsForApproveApply{
        ApplyIds: &listApplyIdsbody,
        ApplyResult: &applyResultOpenApplyIdsForApproveApply,
    }
}
```

```
response, err := client.BatchApproveApply(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	审核申请成功。
400	BadRequest。

## 11.2.3 获取申请详情

### 功能介绍

获取申请详情。此功能仅用作信息详情展示，不用做业务处理，因此不展示编号等后台参数。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/applies/{apply\_id}

表 11-124 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
apply_id	是	String	审核信息ID。

## 请求参数

表 11-125 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-126 响应 Body 参数

参数	参数类型	描述
id	String	申请编号。
api_apply_status	String	申请状态。 枚举值： <ul style="list-style-type: none"> <li>STATUS_TYPE_PENDING_APPROVAL：待审核</li> <li>STATUS_TYPE_REJECTED：已驳回</li> <li>STATUS_TYPE_PENDING_CHECK：待确认</li> <li>STATUS_TYPE_PENDING_EXECUTE：待执行</li> <li>STATUS_TYPE_SYNCHRONOUS_EXECUTE：同步执行</li> <li>STATUS_TYPE_FORCED_CANCEL：强制取消</li> <li>STATUS_TYPE_PASSED：已通过</li> </ul>

参数	参数类型	描述
api_apply_type	String	申请类型。 枚举值： <ul style="list-style-type: none"> <li>• APPLY_TYPE_PUBLISH: 发布API</li> <li>• APPLY_TYPE_AUTHORIZE: API主动授权</li> <li>• APPLY_TYPE_APPLY: 审核API</li> <li>• APPLY_TYPE_RENEW: 申请续约API</li> <li>• APPLY_TYPE_STOP: 申请停用API</li> <li>• APPLY_TYPE_RECOVER: 申请恢复API</li> <li>• APPLY_TYPE_API_CANCEL_AUTHORIZE: API取消授权</li> <li>• APPLY_TYPE_APP_CANCEL_AUTHORIZE: APP取消授权</li> <li>• APPLY_TYPE_OFFLINE: 申请下线</li> </ul>
api_id	String	API编号。
api_name	String	API名称。
api_using_time	Long	使用截止时间。
app_id	String	APP编号。
app_name	String	APP名称。
apply_time	Long	申请时间。
approval_time	Long	授权时间。
approver_name	String	审核人名称。
comment	String	审核评论。
user_name	String	申请人姓名。

**状态码： 400**

**表 11-127 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

获取ID为89374656efeb5e5e8f2cb08ec94cb78a的申请审核记录。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/applies/89374656efeb5e5e8f2cb08ec94cb78a
```

## 响应示例

**状态码： 200**

Success。

```
{
  "id" : null,
  "api_apply_status" : "STATUS_TYPE_PASSED",
  "api_apply_type" : "APPLY_TYPE_APPLY",
  "api_id" : null,
  "api_name" : "api_demo",
  "api_using_time" : 1580452617000,
  "app_id" : null,
  "app_name" : "app_demo",
  "apply_time" : 1578875421000,
  "approval_time" : null,
  "approver_name" : null,
  "comment" : "通过",
  "user_name" : "张三"
}
```

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.3 消息管理接口

### 11.3.1 查询消息列表

#### 功能介绍

查询审核中心的通知消息列表。与申请不同，通知类消息，无法驳回，仅能在指定的时间范围内作出处理。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/service/messages



表 11-128 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-129 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	查询起始坐标，即跳过前X条数据。仅支持0或limit的整数倍，不满足则向下取整。
limit	否	Integer	查询条数限制。
api_name	否	String	API名称。

## 请求参数

表 11-130 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-131 响应 Body 参数

参数	参数类型	描述
total	Integer	符合条件的申请总数。
records	Array of <a href="#">RecordForApplyDetail</a> objects	本次返回的申请列表。

表 11-132 RecordForApplyDetail

参数	参数类型	描述
id	String	申请编号。
api_apply_status	String	申请状态。 枚举值： <ul style="list-style-type: none"> <li>STATUS_TYPE_PENDING_APPROVAL: 待审核</li> <li>STATUS_TYPE_REJECTED: 已驳回</li> <li>STATUS_TYPE_PENDING_CHECK: 待确认</li> <li>STATUS_TYPE_PENDING_EXECUTE: 待执行</li> <li>STATUS_TYPE_SYNCHRONOUS_EXECUTE: 同步执行</li> <li>STATUS_TYPE_FORCED_CANCEL: 强制取消</li> <li>STATUS_TYPE_PASSED: 已通过</li> </ul>
api_apply_type	String	申请类型。 枚举值： <ul style="list-style-type: none"> <li>APPLY_TYPE_PUBLISH: 发布API</li> <li>APPLY_TYPE_AUTHORIZE: API主动授权</li> <li>APPLY_TYPE_APPLY: 审核API</li> <li>APPLY_TYPE_RENEW: 申请续约API</li> <li>APPLY_TYPE_STOP: 申请停用API</li> <li>APPLY_TYPE_RECOVER: 申请恢复API</li> <li>APPLY_TYPE_API_CANCEL_AUTHORIZE: API取消授权</li> <li>APPLY_TYPE_APP_CANCEL_AUTHORIZE: APP取消授权</li> <li>APPLY_TYPE_OFFLINE: 申请下线</li> </ul>
api_id	String	API编号。
api_name	String	API名称。

参数	参数类型	描述
api_using_time	Long	使用截止时间。
app_id	String	APP编号。
app_name	String	APP名称。
apply_time	Long	申请时间。
approval_time	Long	授权时间。
approver_name	String	审核人名称。
comment	String	审核评论。
user_name	String	申请人姓名。

**状态码： 400**

**表 11-133 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询审核中心的通知消息列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/messages
```

## 响应示例

**状态码： 200**

查询结果成功，符合条件的消息总数。

```
{
  "total": 1,
  "records": [ {
    "id": "a1c93c3cb17d9747fc41df0377b6fe0f",
    "api_apply_status": "STATUS_TYPE_PASSED",
    "api_apply_type": "APPLY_TYPE_APPLY",
    "api_id": "47046fe7830c1be77cb0dc23bd86afa5",
    "api_name": "api_demo",
    "api_using_time": 1580452617000,
    "app_id": "908489209a320df61607355c57c82882",
    "app_name": "app_demo",
    "apply_time": 1578875421000,
    "approval_time": 1578897421000,
    "approver_name": "李四",
    "comment": "通过",
```

```
"user_name" : "张三"  
}]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListMessageSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListMessageRequest request = new ListMessageRequest();  
        try {  
            ListMessageResponse response = client.listMessage(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListMessageRequest()
        response = client.list_message(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListMessageRequest{}
    response, err := client.ListMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```

```

} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询结果成功，符合条件的消息总数。
400	BadRequest。

## 11.3.2 处理消息

### 功能介绍

对收到的通知消息进行确认，可以在指定的时间范围内选择何时进行处理。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/messages

表 11-134 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 11-135 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定 SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有 Body体的情况下必选，没有 Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-136 请求 Body 参数

参数	是否必选	参数类型	描述
message_id	否	String	消息编号。
action	否	Integer	执行动作。0=立刻执行, 1=定期执行。
time	否	String	使用截止时间。仅定期执行需要此参数，默认服务器当前时间三天后。

## 响应参数

状态码： 400

表 11-137 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

对收到的通知消息进行确认。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/messages
{
  "message_id": "c6c1229f3387848b6bf69eec7004a60",
  "action": 0,
```

```
"time" : "2022-03-23T10:00:00.000Z"  
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

对收到的通知消息进行确认。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ConfirmMessageSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))  
            .build();  
        ConfirmMessageRequest request = new ConfirmMessageRequest();  
        OpenApiParaForCheckMessage body = new OpenApiParaForCheckMessage();  
        body.withTime("2022-03-23T10:00:00.000Z");  
        body.withAction(0);  
        body.withMessageId("c6c1229f3387848b6fb69eec7004a60");  
        request.withBody(body);  
        try {  
            ConfirmMessageResponse response = client.confirmMessage(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
        }  
    }  
}
```



```
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

对收到的通知消息进行确认。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ConfirmMessageRequest()
        request.body = OpenApiParaForCheckMessage(
            time="2022-03-23T10:00:00.000Z",
            action=0,
            message_id="c6c1229f3387848b6bfb69eec7004a60"
        )
        response = client.confirm_message(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

对收到的通知消息进行确认。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ConfirmMessageRequest{
    timeOpenApiParaForCheckMessage:= "2022-03-23T10:00:00.000Z"
    actionOpenApiParaForCheckMessage:= int32(0)
    messageIdOpenApiParaForCheckMessage:= "c6c1229f3387848b6bfb69eec7004a60"
    request.Body = &model.OpenApiParaForCheckMessage{
        Time: &timeOpenApiParaForCheckMessage,
        Action: &actionOpenApiParaForCheckMessage,
        MessageId: &messageIdOpenApiParaForCheckMessage,
    }
}
response, err := client.ConfirmMessage(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	确认消息成功。
400	BadRequest。

## 11.3.3 获取消息详情

### 功能介绍

获取消息详情。此功能仅用作信息详情展示，不用做业务处理，因此不展示编号等后台参数。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/service/messages/{message\_id}

表 11-138 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
message_id	是	String	消息信息ID。

## 请求参数

表 11-139 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-140 响应 Body 参数

参数	参数类型	描述
id	String	申请编号。

参数	参数类型	描述
api_apply_status	String	申请状态。 枚举值： <ul style="list-style-type: none"> <li>STATUS_TYPE_PENDING_APPROVAL: 待审核</li> <li>STATUS_TYPE_REJECTED: 已驳回</li> <li>STATUS_TYPE_PENDING_CHECK: 待确认</li> <li>STATUS_TYPE_PENDING_EXECUTE: 待执行</li> <li>STATUS_TYPE_SYNCHRONOUS_EXECUTE: 同步执行</li> <li>STATUS_TYPE_FORCED_CANCEL: 强制取消</li> <li>STATUS_TYPE_PASSED: 已通过</li> </ul>
api_apply_type	String	申请类型。 枚举值： <ul style="list-style-type: none"> <li>APPLY_TYPE_PUBLISH: 发布API</li> <li>APPLY_TYPE_AUTHORIZE: API主动授权</li> <li>APPLY_TYPE_APPLY: 审核API</li> <li>APPLY_TYPE_RENEW: 申请续约API</li> <li>APPLY_TYPE_STOP: 申请停用API</li> <li>APPLY_TYPE_RECOVER: 申请恢复API</li> <li>APPLY_TYPE_API_CANCEL_AUTHORIZE: API取消授权</li> <li>APPLY_TYPE_APP_CANCEL_AUTHORIZE: APP取消授权</li> <li>APPLY_TYPE_OFFLINE: 申请下线</li> </ul>
api_id	String	API编号。
api_name	String	API名称。
api_using_time	Long	使用截止时间。
app_id	String	APP编号。
app_name	String	APP名称。
apply_time	Long	申请时间。
approval_time	Long	授权时间。
approver_name	String	审核人名称。
comment	String	审核评论。
user_name	String	申请人姓名。

**状态码： 400**

**表 11-141 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

获取ID为51159105c7838353d2834181d978af50的审核通知消息。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/messages/51159105c7838353d2834181d978af50
```

## 响应示例

**状态码： 200**

Success。

```
{
  "id" : null,
  "api_apply_status" : "STATUS_TYPE_PASSED",
  "api_apply_type" : "APPLY_TYPE_APPLY",
  "api_id" : null,
  "api_name" : "api_demo",
  "api_using_time" : 1580452617000,
  "app_id" : null,
  "app_name" : "app_demo",
  "apply_time" : 1578875421000,
  "approval_time" : null,
  "approver_name" : null,
  "comment" : "通过",
  "user_name" : "张三"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowMessageDetailSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowMessageDetailRequest request = new ShowMessageDetailRequest();
    request.withMessageId("{message_id}");
    try {
        ShowMessageDetailResponse response = client.showMessageDetail(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMessageDetailRequest()
        request.message_id = "{message_id}"
        response = client.show_message_detail(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMessageDetailRequest{}
    request.MessageId = "{message_id}"
    response, err := client.ShowMessageDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.4 授权管理接口

### 11.4.1 查询 API 已授权的 APP

#### 功能介绍

查询API已授权的APP。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/service/authorize/apis/{api\_id}

表 11-142 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。

表 11-143 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或LIMIT的整数倍, 不满足则向下取整。
limit	否	Integer	查询条数限制。

#### 请求参数

表 11-144 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。



参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-145 响应 Body 参数

参数	参数类型	描述
total	Integer	符合条件的APP总数。
records	Array of <a href="#">RecordForGetAuthApp</a> objects	本次返回的APP列表。

表 11-146 RecordForGetAuthApp

参数	参数类型	描述
app_id	String	应用ID编号。
app_name	String	应用名称。
instance_id	String	集群实例id。
instance_name	String	集群实例名称。
api_using_time	Long	使用截止时间。
approval_time	Long	授权时间。

参数	参数类型	描述
relationship_type	String	绑定关系。 枚举值： <ul style="list-style-type: none"> <li>LINK_WAITING_CHECK: 待审核</li> <li>LINKED: 已绑定</li> <li>OFFLINE_WAITING_CHECK: 待下线</li> <li>RENEW_WAITING_CHECK: 待续约</li> </ul>
static_params	Array of <b>StaticParam</b> objects	静态参数列表。

表 11-147 StaticParam

参数	参数类型	描述
para_name	String	静态参数名。
para_value	String	静态参数值。

状态码： 400

表 11-148 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询API已授权的APP。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/authorize/apis/
```

## 响应示例

状态码： 200

查询结果成功，API已授权的APP列表。

```
{
  "total": 1,
  "records": [{
    "app_id": "908489209a320df61607355c57c82882",
    "app_name": "app_demo",
    "api_using_time": 1580452617000,
    "approval_time": 1578897421000
  }]
}
```

```
    }]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class SearchAuthorizeAppSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        SearchAuthorizeAppRequest request = new SearchAuthorizeAppRequest();  
        request.withApiId("{api_id}");  
        try {  
            SearchAuthorizeAppResponse response = client.searchAuthorizeApp(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchAuthorizeAppRequest()
        request.api_id = "{api_id}"
        response = client.search_authorize_app(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchAuthorizeAppRequest{}
    request.ApiId = "{api_id}"
    response, err := client.SearchAuthorizeApp(request)
```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询结果成功，API已授权的APP列表。
400	BadRequest。

## 11.4.2 查询 APP 已拥有授权的 API

### 功能介绍

查询APP已拥有授权的API。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/authorize/apps/{app\_id}

表 11-149 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
app_id	是	String	APP编号。

表 11-150 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	查询起始坐标，即跳过前X条数据。仅支持0或LIMIT的整数倍，不满足则向下取整。

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。

## 请求参数

表 11-151 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-152 响应 Body 参数

参数	参数类型	描述
total	Integer	应用已绑定的API总个数。
records	Array of <a href="#">AppBindApilnfo</a> objects	应用已绑定的API列表。

表 11-153 AppBindApiInfo

参数	参数类型	描述
id	String	API ID。
name	String	API名称。
description	String	API描述。
approval_time	Long	审核时间。
manager	String	API 审核人名称。
deadline	Long	使用截止时间。
relationship_type	String	绑定关系。 枚举值： <ul style="list-style-type: none"> <li>LINK_WAITING_CHECK: 待审核</li> <li>LINKED: 已绑定</li> <li>OFFLINE_WAITING_CHECK: 待下线</li> <li>RENEW_WAITING_CHECK: 待续约</li> </ul>
static_params	Array of <a href="#">StaticParam</a> objects	静态参数列表。

表 11-154 StaticParam

参数	参数类型	描述
para_name	String	静态参数名。
para_value	String	静态参数值。

状态码： 400

表 11-155 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询APP已拥有授权的API。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/authorize/apps/760715eb1bfce0c575abab3be3bd41e6
```

## 响应示例

状态码： 200

Success。

```
{
  "total" : 1,
  "records" : [ {
    "id" : "2342936b7862317d0a81bfcfb657f243",
    "name" : "testApi",
    "description" : "test api",
    "approval_time" : 1578897421000,
    "manager" : "admin",
    "deadline" : 1580452617000
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchBindApiSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        SearchBindApiRequest request = new SearchBindApiRequest();
        request.withAppId("{app_id}");
        try {
            SearchBindApiResponse response = client.searchBindApi(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```



```
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchBindApiRequest()
        request.app_id = "{app_id}"
        response = client.search_bind_api(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.SearchBindApiRequest{}
request.AppId = "{app_id}"
response, err := client.SearchBindApi(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

# 11.5 服务目录管理接口

## 11.5.1 获取当前目录下的所有类型列表

### 功能介绍

获取当前目录下所有类型列表（包括API和目录，均以目录的数据格式形式展示）。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/servicecatalogs/{catalog\_id}/detail

表 11-156 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
catalog_id	是	String	目录ID编号。

表 11-157 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	查询起始坐标, 即跳过前X条数据。
limit	否	Integer	查询条数限制。

## 请求参数

表 11-158 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token, 使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型, 指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型(格式), 有Body体的情况下必选, 没有Body体无需填写。如果请求消息体中含有中文字符, 则需要通过charset=utf8指定中文字符集, 例如取值为: application/json;charset=utf8。

## 响应参数

状态码: 200

表 11-159 响应 Body 参数

参数	参数类型	描述
total	Integer	符合条件的数据总数。
api_catalogs	Array of <a href="#">RecordForGetAllCatalog</a> objects	本次返回的APP列表。

表 11-160 RecordForGetAllCatalog

参数	参数类型	描述
catalog_id	String	目录ID编号。
pid	String	父目录编号。
name	String	名称。
description	String	目录描述。
api_catalog_type	String	目录资源类型。 枚举值： <ul style="list-style-type: none"> <li>• CATALOG: 目录</li> <li>• API: API</li> </ul>
create_time	Long	创建时间。
create_user	String	创建者。
update_time	Long	更新时间。
update_user	String	更新者。

状态码： 400

表 11-161 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

获取目录ID为51159105c7838353d2834181d978af50下的目录列表与API列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/51159105c7838353d2834181d978af50/detail
```

## 响应示例

状态码： 200

Success。

```
{
  "total": 1,
  "api_catalogs": [ {
    "catalog_id": "2847131b4d5a26c5bd4bfd9d8f63f577",
    "pid": "8c9850726a1ee050be2b67e43c688870",
    "name": "demo",
    "description": "this is a demo",
    "api_catalog_type": "CATALOG",
    "create_time": 1578284788000,
    "create_user": "张三",
    "update_time": 1578284788000,
    "update_user": "张三"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListAllCatalogListSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAllCatalogListRequest request = new ListAllCatalogListRequest();
        request.withCatalogId("{catalog_id}");
        try {
            ListAllCatalogListResponse response = client.listAllCatalogList(request);
        }
    }
}
```

```

        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
}

```

## Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAllCatalogListRequest()
        request.catalog_id = "{catalog_id}"
        response = client.list_all_catalog_list(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.

```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAllCatalogListRequest{}
request.CatalogId = "{catalog_id}"
response, err := client.ListAllCatalogList(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.2 获取当前目录下的 API 列表

### 功能介绍

获取当前目录下的API列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/servicecatalogs/{catalog\_id}/apis

表 11-162 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。
catalog_id	是	String	目录ID编号。

表 11-163 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	查询起始坐标, 即跳过前X条数据。
limit	否	Integer	查询条数限制。

## 请求参数

表 11-164 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token, 使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型, 指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型(格式), 有Body体的情况下必选, 没有Body体无需填写。如果请求消息体中含有中文字符, 则需要通过charset=utf8指定中文字符集, 例如取值为: application/json;charset=utf8。

## 响应参数

状态码: 200



表 11-165 响应 Body 参数

参数	参数类型	描述
total	Integer	API总数量。
apis	Array of <a href="#">ApiOverview</a> objects	API列表。

表 11-166 ApiOverview

参数	参数类型	描述
id	String	API ID。
name	String	API名称。
group_id	String	API分组ID（共享版）。
description	String	API描述。
status	String	API的状态（共享版）。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_CREATED: API创建状态</li> <li>• API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>• API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>• API_STATUS_PUBLISHED: 已发布状态</li> <li>• API_STATUS_WAITING_STOP: 停用审核状态</li> <li>• API_STATUS_STOPPED: 已停用状态</li> <li>• API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>• API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>• API_STATUS_OFFLINE: 已下线</li> <li>• API_STATUS_OFFLINE_WAIT_REVIEW: 下线待审核状态</li> </ul>
debug_status	String	API调试状态（共享版）。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING: 等待调试</li> <li>• API_DEBUG_FAILED: 调试失败</li> <li>• API_DEBUG_SUCCESS: 调试成功</li> </ul>
publish_messages	Array of <a href="#">ApiPublishDTO</a> objects	发布信息列表（专享版）。

参数	参数类型	描述
type	String	API类型。 枚举值： <ul style="list-style-type: none"> <li>• API_SPECIFIC_TYPE_CONFIGURATION: 配置类API</li> <li>• API_SPECIFIC_TYPE_SCRIPT: 脚本类API</li> <li>• API_SPECIFIC_TYPE_REGISTER: 注册类API</li> </ul>
manager	String	API审核人。
create_user	String	API创建者。
create_time	Long	API创建时间。
authorization_status	String	授权状态。 枚举值： <ul style="list-style-type: none"> <li>• NO_AUTHORIZATION_REQUIRED: 无需授权</li> <li>• UNAUTHORIZED: 未授权</li> <li>• AUTHORIZED: 已授权</li> </ul>

表 11-167 ApiPublishDTO

参数	参数类型	描述
id	String	发布编号。
api_id	String	API编号。
instance_id	String	集群ID编号。
instance_name	String	集群名称。

参数	参数类型	描述
api_status	String	API的状态。 枚举值： <ul style="list-style-type: none"> <li>• API_STATUS_CREATED: API创建状态</li> <li>• API_STATUS_PUBLISH_WAIT_REVIEW: 发布审核状态</li> <li>• API_STATUS_PUBLISH_REJECT: 驳回状态</li> <li>• API_STATUS_PUBLISHED: 已发布状态</li> <li>• API_STATUS_WAITING_STOP: 停用审核状态</li> <li>• API_STATUS_STOPPED: 已停用状态</li> <li>• API_STATUS_RECOVER_WAIT_REVIEW: 恢复审核状态</li> <li>• API_STATUS_WAITING_OFFLINE: 下线审核状态</li> <li>• API_STATUS_OFFLINE: 已下线</li> </ul>
api_debug	String	API调试状态。 枚举值： <ul style="list-style-type: none"> <li>• API_DEBUG_WAITING: 等待调试</li> <li>• API_DEBUG_FAILED: 调试失败</li> <li>• API_DEBUG_SUCCESS: 调试成功</li> </ul>

状态码： 400

表 11-168 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

获取目录ID为51159105c7838353d2834181d978af50下的API列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/51159105c7838353d2834181d978af50/apis
```

## 响应示例

状态码： 200

Success。

```
{
  "total": 1,
```

```
"apis" : [ {  
  "id" : "fd0beac79418d65e3b3ed24a6e33b24b",  
  "name" : "testApi",  
  "create_time" : 1579162215000,  
  "status" : "API_STATUS_CREATED",  
  "manager" : "admin",  
  "type" : null,  
  "debug_status" : "API_DEBUG_WAITING"  
} ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListApiCatalogListSolution {  
  public static void main(String[] args) {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    // environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running  
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    String ak = System.getenv("CLOUD_SDK_AK");  
    String sk = System.getenv("CLOUD_SDK_SK");  
    String projectId = "{project_id}";  
  
    ICredential auth = new BasicCredentials()  
      .withProjectId(projectId)  
      .withAk(ak)  
      .withSk(sk);  
  
    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
      .withCredential(auth)  
      .withRegion(DataArtsStudioRegion.valueOf("<YOUR_REGION>"))  
      .build();  
    ListApiCatalogListRequest request = new ListApiCatalogListRequest();  
    request.withCatalogId("{catalog_id}");  
    try {  
      ListApiCatalogListResponse response = client.listApiCatalogList(request);  
      System.out.println(response.toString());  
    } catch (ConnectionException e) {  
      e.printStackTrace();  
    } catch (RequestTimeoutException e) {  
      e.printStackTrace();  
    } catch (ServiceResponseException e) {  
      e.printStackTrace();  
      System.out.println(e.getHttpStatusCode());  
      System.out.println(e.getRequestId());  
      System.out.println(e.getErrorCode());  
      System.out.println(e.getErrorMsg());  
    }  
  }  
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListApiCatalogListRequest()
        request.catalog_id = "{catalog_id}"
        response = client.list_api_catalog_list(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```

        WithCredential(auth).
        Build()

    request := &model.ListApiCatalogListRequest{}
    request.CatalogId = "{catalog_id}"
    response, err := client.ListApiCatalogList(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.3 获取当前目录下的目录列表（全量）

### 功能介绍

获取当前目录下的目录列表（全量数据，不分页，推荐仅用于生成目录树等无法分页的场景）。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/servicecatalogs/{catalog\_id}/catalogs

表 11-169 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
catalog_id	是	String	目录ID编号。

表 11-170 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或LIMIT的整数倍, 不满足则向下取整。

## 请求参数

表 11-171 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token, 使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型, 指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型(格式), 有Body体的情况下必选, 没有Body体无需填写。如果请求消息体中含有中文字符, 则需要通过charset=utf8指定中文字符集, 例如取值为: application/json;charset=utf8。

## 响应参数

状态码: 200

表 11-172 响应 Body 参数

参数	参数类型	描述
total	Integer	符合条件的数据总数。
catalogs	Array of <a href="#">RecordForGetAllCatalog</a> objects	本次返回的APP列表。

表 11-173 RecordForGetAllCatalog

参数	参数类型	描述
catalog_id	String	目录ID编号。
pid	String	父目录编号。
name	String	名称。
description	String	目录描述。
api_catalog_type	String	目录资源类型。 枚举值： <ul style="list-style-type: none"> <li>• CATALOG: 目录</li> <li>• API: API</li> </ul>
create_time	Long	创建时间。
create_user	String	创建者。
update_time	Long	更新时间。
update_user	String	更新者。

状态码： 400

表 11-174 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

获取目录id为51159105c7838353d2834181d978af50下的目录列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/51159105c7838353d2834181d978af50/catalogs
```

## 响应示例

状态码： 200

Success。

```
{
  "total": 1,
  "catalogs": [ {
    "catalog_id": "2847131b4d5a26c5bd4bfd9d8f63f577",
    "pid": "8c9850726a1ee050be2b67e43c688870",
    "name": "demo",
    "description": "this is a demo",
```



```
"api_catalog_type": "CATALOG",  
"create_time": 1578284788000,  
"create_user": "张三",  
"update_time": 1578284788000,  
"update_user": "张三"  
} ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListCatalogListSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListCatalogListRequest request = new ListCatalogListRequest();  
        request.withCatalogId("{catalog_id}");  
        try {  
            ListCatalogListResponse response = client.listCatalogList(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCatalogListRequest()
        request.catalog_id = "{catalog_id}"
        response = client.list_catalog_list(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).  
Build())  
  
request := &model.ListCatalogListRequest{}  
request.CatalogId = "{catalog_id}"  
response, err := client.ListCatalogList(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.4 更新服务目录

### 功能介绍

更新服务目录。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/service/servicecatalogs/{catalog\_id}

表 11-175 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
catalog_id	是	String	目录ID编号。

## 请求参数

表 11-176 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID</a> 和 <a href="#">工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-177 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	名称。
description	否	String	描述。

## 响应参数

状态码： 200

表 11-178 响应 Body 参数

参数	参数类型	描述
catalog_id	String	目录ID编号。
name	String	名称。
description	String	描述。
create_time	Long	创建时间。
create_user	String	创建者。

参数	参数类型	描述
update_time	Long	更新时间。
update_user	String	更新者。

状态码： 400

表 11-179 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

更新ID为51159105c7838353d2834181d978af50的目录信息。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/51159105c7838353d2834181d978af50
{
  "name": "demo",
  "description": "this is a demo"
}
```

## 响应示例

状态码： 200

Success。

```
{
  "catalog_id": "2847131b4d5a26c5bd4bfd9d8f63f577",
  "name": "demo",
  "description": "this is a demo",
  "create_time": 1578284788000,
  "create_user": "张三",
  "update_time": 1578284788000,
  "update_user": "张三"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新ID为51159105c7838353d2834181d978af50的目录信息。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateCatalogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateCatalogRequest request = new UpdateCatalogRequest();
        request.withCatalogId("{catalog_id}");
        ApiCatalogUpdateParaDTO body = new ApiCatalogUpdateParaDTO();
        body.withDescription("this is a demo");
        body.withName("demo");
        request.withBody(body);
        try {
            UpdateCatalogResponse response = client.updateCatalog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

更新ID为51159105c7838353d2834181d978af50的目录信息。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateCatalogRequest()
    request.catalog_id = "{catalog_id}"
    request.body = ApiCatalogUpdateParaDTO(
        description="this is a demo",
        name="demo"
    )
    response = client.update_catalog(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

更新ID为51159105c7838353d2834181d978af50的目录信息。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateCatalogRequest{}
    request.CatalogId = "{catalog_id}"
    descriptionApiCatalogUpdateParaDto := "this is a demo"
    nameApiCatalogUpdateParaDto := "demo"
    request.Body = &model.ApiCatalogUpdateParaDto{
        Description: &descriptionApiCatalogUpdateParaDto,
        Name: &nameApiCatalogUpdateParaDto,
    }
```

```
}  
response, err := client.UpdateCatalog(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.5 查询服务目录

### 功能介绍

查询服务目录。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/servicecatalogs/{catalog\_id}

表 11-180 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
catalog_id	是	String	目录ID编号。



## 请求参数

表 11-181 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-182 响应 Body 参数

参数	参数类型	描述
catalog_id	String	目录ID编号。
name	String	名称。
path	String	路径。
catalog_total	Integer	子目录数。
api_total	Integer	API数。
description	String	描述。
create_time	Long	创建时间。
create_user	String	创建者。
update_time	Long	更新时间。
update_user	String	更新者。

**状态码： 400**

**表 11-183 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询ID为51159105c7838353d2834181d978af50服务目录。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/51159105c7838353d2834181d978af50
```

## 响应示例

**状态码： 200**

Success。

```
{
  "catalog_id": "2847131b4d5a26c5bd4bfd9d8f63f577",
  "name": "demo",
  "path": "/demo1/demo2/demo",
  "catalog_total": 1,
  "api_total": 1,
  "description": "this is a demo",
  "create_time": 1578284788000,
  "create_user": "张三",
  "update_time": 1578284788000,
  "update_user": "张三"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowCatalogDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    }
}
```

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowCatalogDetailRequest request = new ShowCatalogDetailRequest();
request.withCatalogId("{catalog_id}");
try {
    ShowCatalogDetailResponse response = client.showCatalogDetail(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCatalogDetailRequest()
        request.catalog_id = "{catalog_id}"
        response = client.show_catalog_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowCatalogDetailRequest{}
    request.CatalogId = "{catalog_id}"
    response, err := client.ShowCatalogDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.6 创建服务目录

### 功能介绍

创建服务目录。根目录编号为0。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/service/servicecatalogs

表 11-184 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 11-185 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-186 请求 Body 参数

参数	是否必选	参数类型	描述
pid	否	String	父目录编号。
name	否	String	名称。
description	否	String	描述。

## 响应参数

状态码： 200

表 11-187 响应 Body 参数

参数	参数类型	描述
catalog_id	String	目录ID编号。
name	String	名称。
description	String	描述。
create_time	Long	创建时间。
create_user	String	创建者。
update_time	Long	更新时间。
update_user	String	更新者。

状态码： 400

表 11-188 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

创建服务目录。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs  
  
{  
  "pid": "0",  
  "name": "demo111",  
  "description": "创建目录的备注。"  
}
```

## 响应示例

状态码： 200

Success。

```
{  
  "catalog_id": "2847131b4d5a26c5bd4bfd9d8f63f577",  
  "name": "demo",  
  "description": "this is a demo",  
  "create_time": 1578284788000,  
  "create_user": "张三",  
}
```

```
"update_time" : 1578284788000,  
"update_user" : "张三"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建服务目录。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class CreateServiceCatalogSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateServiceCatalogRequest request = new CreateServiceCatalogRequest();  
        ApiCatalogCreateParaDTO body = new ApiCatalogCreateParaDTO();  
        body.withDescription("创建目录的备注。");  
        body.withName("demo111");  
        body.withPid("0");  
        request.withBody(body);  
        try {  
            CreateServiceCatalogResponse response = client.createServiceCatalog(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

## Python

创建服务目录。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateServiceCatalogRequest()
        request.body = ApiCatalogCreateParaDTO(
            description="创建目录的备注。",
            name="demo111",
            pid="0"
        )
        response = client.create_service_catalog(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

创建服务目录。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```



```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateServiceCatalogRequest{
    descriptionApiCatalogCreateParaDto:= "创建目录的备注。"
    nameApiCatalogCreateParaDto:= "demo111"
    pidApiCatalogCreateParaDto:= "0"
    request.Body = &model.ApiCatalogCreateParaDto{
        Description: &descriptionApiCatalogCreateParaDto,
        Name: &nameApiCatalogCreateParaDto,
        Pid: &pidApiCatalogCreateParaDto,
    }
}
response, err := client.CreateServiceCatalog(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.7 批量删除目录

### 功能介绍

批量删除服务目录。删除目录的同时会删除其下的所有子目录，不支持删除带有API的目录。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/servicecatalogs/batch-delete

表 11-189 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 11-190 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-191 请求 Body 参数

参数	是否必选	参数类型	描述
ids	否	Array of strings	目录ID编号。

## 响应参数

状态码： 400

表 11-192 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。

参数	参数类型	描述
error_msg	String	错误描述。

## 请求示例

批量删除服务目录。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/batch-delete
{
  "ids" : [ "a8523d5a41e7c6c1b38c2bc0984f097e" ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

批量删除服务目录。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteServiceCatalogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
```

```
DeleteServiceCatalogRequest request = new DeleteServiceCatalogRequest();
ApiCatalogDeleteParaDTO body = new ApiCatalogDeleteParaDTO();
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("a8523d5a41e7c6c1b38c2bc0984f097e");
body.withIds(listbodyIds);
request.withBody(body);
try {
    DeleteServiceCatalogResponse response = client.deleteServiceCatalog(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

批量删除服务目录。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteServiceCatalogRequest()
        listbody = [
            "a8523d5a41e7c6c1b38c2bc0984f097e"
        ]
        request.body = ApiCatalogDeleteParaDTO(
            ids=listbody
        )
        response = client.delete_service_catalog(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

批量删除服务目录。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteServiceCatalogRequest{}
    var listIdsbody = []string{
        "a8523d5a41e7c6c1b38c2bc0984f097e",
    }
    request.Body = &model.ApiCatalogDeleteParaDto{
        Ids: &listIdsbody,
    }
    response, err := client.DeleteServiceCatalog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	Success。
400	BadRequest。

## 11.5.8 移动当前目录至新目录

### 功能介绍

移动当前目录至新目录。移动目录的同时会移动其下的所有子目录与api。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/servicecatalogs/{catalog\_id}/move

表 11-193 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
catalog_id	是	String	目录ID编号。

### 请求参数

表 11-194 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-195 请求 Body 参数

参数	是否必选	参数类型	描述
target_pid	否	String	父目录ID编号。

## 响应参数

状态码： 400

表 11-196 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

移动当前目录至新目录。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/51159105c7838353d2834181d978af50/move
{
  "target_pid" : "47d0c3723fd4bd3fdcffbcf4ec40c693"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

移动当前目录至新目录。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class MigrateCatalogSolution {
    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
MigrateCatalogRequest request = new MigrateCatalogRequest();
request.withCatalogId("{catalog_id}");
CatalogMoveParaDTO body = new CatalogMoveParaDTO();
body.withTargetPid("47d0c3723fd4bd3fdcffbcf4ec40c693");
request.withBody(body);
try {
    MigrateCatalogResponse response = client.migrateCatalog(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

移动当前目录至新目录。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()
```



```
try:
    request = MigrateCatalogRequest()
    request.catalog_id = "{catalog_id}"
    request.body = CatalogMoveParaDTO(
        target_pid="47d0c3723fd4bd3fdcffbcf4ec40c693"
    )
    response = client.migrate_catalog(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

移动当前目录至新目录。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.MigrateCatalogRequest{}
    request.CatalogId = "{catalog_id}"
    targetPidCatalogMoveParaDto := "47d0c3723fd4bd3fdcffbcf4ec40c693"
    request.Body = &model.CatalogMoveParaDto{
        TargetPid: &targetPidCatalogMoveParaDto,
    }
    response, err := client.MigrateCatalog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	Success。
400	BadRequest。

## 11.5.9 批量移动 API 至新目录

### 功能介绍

批量移动API至新目录。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/servicecatalogs/apis/batch-move

表 11-197 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 11-198 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。

参数	是否必选	参数类型	描述
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-199 请求 Body 参数

参数	是否必选	参数类型	描述
target_pid	否	String	父目录ID编号。
apis	否	Array of strings	需要移动的目录。

## 响应参数

状态码： 400

表 11-200 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

批量移动API至新目录。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/apis/batch-move
{
  "target_pid": "569e65bf4318b947ce5246e9d03c47e7",
  "apis": [ "43ad1364493d7c04c8e09ae98d877527" ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

批量移动API至新目录。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class MigrateApiSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        MigrateApiRequest request = new MigrateApiRequest();
        ApiMoveParaDTO body = new ApiMoveParaDTO();
        List<String> listbodyApis = new ArrayList<>();
        listbodyApis.add("43ad1364493d7c04c8e09ae98d877527");
        body.withApis(listbodyApis);
        body.withTargetPid("569e65bf4318b947ce5246e9d03c47e7");
        request.withBody(body);
        try {
            MigrateApiResponse response = client.migrateApi(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

批量移动API至新目录。

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = MigrateApiRequest()
        listApisbody = [
            "43ad1364493d7c04c8e09ae98d877527"
        ]
        request.body = ApiMoveParaDTO(
            apis=listApisbody,
            target_pid="569e65bf4318b947ce5246e9d03c47e7"
        )
        response = client.migrate_api(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

批量移动API至新目录。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```
client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.MigrateApiRequest{}
var listApisbody = []string{
    "43ad1364493d7c04c8e09ae98d877527",
}
targetPidApiMoveParaDto:= "569e65bf4318b947ce5246e9d03c47e7"
request.Body = &model.ApiMoveParaDto{
    Apis: &listApisbody,
    TargetPid: &targetPidApiMoveParaDto,
}
response, err := client.MigrateApi(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	Success。
400	BadRequest。

## 11.5.10 通过路径获取 ID

### 功能介绍

通过路径获取ID。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/servicecatalogs/ids

表 11-201 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-202 Query 参数

参数	是否必选	参数类型	描述
path	是	String	路径。
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或LIMIT的整数倍, 不满足则向下取整。

## 请求参数

表 11-203 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-204 响应 Body 参数

参数	参数类型	描述
catalog_id	String	目录ID编号。

状态码： 400

表 11-205 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

通过路径获取ID。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/ids
```

## 响应示例

状态码： 200

Success。

```
{
  "catalog_id" : "2847131b4d5a26c5bd4bfd9d8f63f577"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class SearchIdByPathSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
```



this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
SearchIdByPathRequest request = new SearchIdByPathRequest();
try {
    SearchIdByPathResponse response = client.searchIdByPath(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SearchIdByPathRequest()
        response = client.search_id_by_path(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchIdByPathRequest{}
    response, err := client.SearchIdByPath(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.11 通过 ID 获取路径

### 功能介绍

通过ID获取路径。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/service/servicecatalogs/{catalog\_id}/paths

表 11-206 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
catalog_id	是	String	目录ID编号。

表 11-207 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标，即跳过前X条数据。仅支持0或LIMIT的整数倍，不满足则向下取整。

## 请求参数

表 11-208 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。

参数	是否必选	参数类型	描述
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-209 响应 Body 参数

参数	参数类型	描述
path	String	路径。

状态码： 400

表 11-210 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

通过ID获取路径。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/51159105c7838353d2834181d978af50/paths
```

## 响应示例

状态码： 200

Success。

```
{
  "path" : "/demo1/demo2/demo"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowPathByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowPathByIdRequest request = new ShowPathByIdRequest();
        request.withCatalogId("{catalog_id}");
        try {
            ShowPathByIdResponse response = client.showPathById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowPathByIdRequest()
    request.catalog_id = "{catalog_id}"
    response = client.show_path_by_id(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPathByIdRequest{}
    request.CatalogId = "{catalog_id}"
    response, err := client.ShowPathById(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.12 通过目录 ID 获取路径对象

### 功能介绍

通过目录id获取路径对象。通过目录id获取从根目录至当前目录链路上每一层的路径信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/servicecatalogs/{catalog\_id}/layerpaths

表 11-211 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
catalog_id	是	String	目录ID编号。

表 11-212 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标，即跳过前X条数据。仅支持0或LIMIT的整数倍，不满足则向下取整。

## 请求参数

表 11-213 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-214 响应 Body 参数

参数	参数类型	描述
paths	Array of <a href="#">LayerPath</a> objects	路径对象。

表 11-215 LayerPath

参数	参数类型	描述
catalog_id	String	目录ID编号。
name	String	路径名。
order	Integer	路径层级。

状态码： 400



表 11-216 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

通过目录ID获取路径对象。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/servicecatalogs/51159105c7838353d2834181d978af50/layerpaths
```

## 响应示例

状态码： 200

Success。

```
{
  "paths": [ {
    "catalog_id": "2847131b4d5a26c5bd4bfd9d8f63f577",
    "name": "/demo",
    "order": 3
  }, {
    "catalog_id": "3847131b4d565657g435536568f635342",
    "name": "/demo2",
    "order": 2
  }, {
    "catalog_id": "450aa37131b4d5a26c5bcfd9d8f63j653",
    "name": "/demo1",
    "order": 1
  }, {
    "catalog_id": "0",
    "name": "/",
    "order": 0
  }
]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowPathObjectByIdSolution {
    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowPathObjectByldRequest request = new ShowPathObjectByldRequest();
request.withCatalogId("{catalog_id}");
try {
    ShowPathObjectByldResponse response = client.showPathObjectByld(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowPathObjectByldRequest()
        request.catalog_id = "{catalog_id}"
        response = client.show_path_object_by_id(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPathObjectByIdRequest{}
    request.CatalogId = "{catalog_id}"
    response, err := client.ShowPathObjectById(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.5.13 查询服务目录 API 列表

### 功能介绍

查询服务目录API列表。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/market

表 11-217 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 11-218 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-219 请求 Body 参数

参数	是否必选	参数类型	描述
auth_type	否	String	认证类型。 枚举值： <ul style="list-style-type: none"> <li>APP: app secret认证</li> <li>IAM: IAM token认证</li> <li>NONE: 无认证</li> </ul>
visibility	否	String	API可见性。 枚举值： <ul style="list-style-type: none"> <li>WORKSPACE: 工作空间可见</li> <li>PROJECT: 项目可见</li> <li>DOMAIN: 租户可见</li> <li>SPECIFIC_PROJECT: 指定项目可见</li> </ul>
market_sort_type	否	String	排序参数。 枚举值： <ul style="list-style-type: none"> <li>updateTime: 更新时间</li> <li>createTime: 创建时间</li> </ul>
asc_or_desc	否	String	排序方式。 枚举值： <ul style="list-style-type: none"> <li>asc: 升序</li> <li>desc: 降序</li> </ul>
offset	否	Integer	查询起始坐标。
limit	否	Integer	查询条数。
is_owner	否	Boolean	是否显示拥有者。
is_authorized	否	Boolean	是否显示已被授权。
is_update_recently	否	Boolean	是否显示最近更新。
is_release_recently	否	Boolean	是否显示最近发布。
is_hot_recently	否	Boolean	是否显示热销状态。
success_and_failure_rate	否	Boolean	是否显示7天内成功率与失败率。

## 响应参数

状态码： 200

表 11-220 响应 Body 参数

参数	参数类型	描述
total	Integer	API总条数。
apis	Array of <a href="#">AdvancedMallApiDTO</a> objects	API列表。

表 11-221 AdvancedMallApiDTO

参数	参数类型	描述
id	String	API ID。
name	String	API名称。
auth_type	String	认证类型。 枚举值： <ul style="list-style-type: none"> <li>• APP: app secret认证</li> <li>• IAM: IAM token认证</li> <li>• NONE: 无认证</li> </ul>
application_num	Integer	授权使用的应用数量。
call_num	Integer	被调用量。
user_name	String	创建者。
create_time	Long	创建时间。
update_time	Long	更新时间。
is_owner	Boolean	是否当前空间的API。
is_authorized	Boolean	是否已被授权。
is_update_recently	Boolean	是否最近更新（三天内更新过该API）。
is_release_recently	Boolean	是否新品上市（三天内新发布的API）。
is_hot_recently	Boolean	是否热销产品（三天内有其他空间用户申请该API）。
success_rate	String	7天内调用成功率。

参数	参数类型	描述
failure_rate	String	7天内调用失败率。

状态码： 400

表 11-222 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询服务目录API列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/market
{
  "is_authorized": true,
  "is_hot_recently": true,
  "is_owner": true,
  "is_release_recently": true,
  "is_update_recently": true,
  "limit": 12,
  "offset": 0,
  "success_and_failure_rate": true,
  "visibility": "SPECIFIC_PROJECT"
}
```

## 响应示例

状态码： 200

Success。

```
{
  "apis": [ {
    "application_num": 0,
    "auth_type": "NONE",
    "call_num": 1,
    "create_time": 1704420144000,
    "failure_rate": null,
    "id": "bb110fcffb5f319eb08939e877566d6a",
    "is_authorized": null,
    "is_hot_recently": null,
    "is_owner": null,
    "is_release_recently": null,
    "is_update_recently": null,
    "name": "bug_dws",
    "success_rate": null,
    "update_time": 1704532548000,
    "user_name": "user"
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询服务目录API列表。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDataServiceMarketApisSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDataServiceMarketApisRequest request = new ListDataServiceMarketApisRequest();
        MallParaDTO body = new MallParaDTO();
        body.withSuccessAndFailureRate(true);
        body.withIsHotRecently(true);
        body.withIsReleaseRecently(true);
        body.withIsUpdateRecently(true);
        body.withIsAuthorized(true);
        body.withIsOwner(true);
        body.withLimit(12);
        body.withOffset(0);
        body.withVisibility(MallParaDTO.VisibilityEnum.fromValue("SPECIFIC_PROJECT"));
        request.withBody(body);
        try {
            ListDataServiceMarketApisResponse response = client.listDataServiceMarketApis(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```



```
}  
}
```

## Python

查询服务目录API列表。

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListDataServiceMarketApisRequest()  
        request.body = MallParaDTO(  
            success_and_failure_rate=True,  
            is_hot_recently=True,  
            is_release_recently=True,  
            is_update_recently=True,  
            is_authorized=True,  
            is_owner=True,  
            limit=12,  
            offset=0,  
            visibility="SPECIFIC_PROJECT"  
        )  
        response = client.list_data_service_market_apis(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

查询服务目录API列表。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListDataServiceMarketApisRequest{
    successAndFailureRateMallParaDto:= true
    isHotRecentlyMallParaDto:= true
    isReleaseRecentlyMallParaDto:= true
    isUpdateRecentlyMallParaDto:= true
    isAuthorizedMallParaDto:= true
    isOwnerMallParaDto:= true
    limitMallParaDto:= int32(12)
    offsetMallParaDto:= int32(0)
    visibilityMallParaDto:= model.GetMallParaDtoVisibilityEnum().SPECIFIC_PROJECT
    request.Body = &model.MallParaDto{
        SuccessAndFailureRate: &successAndFailureRateMallParaDto,
        IsHotRecently: &isHotRecentlyMallParaDto,
        IsReleaseRecently: &isReleaseRecentlyMallParaDto,
        IsUpdateRecently: &isUpdateRecentlyMallParaDto,
        IsAuthorized: &isAuthorizedMallParaDto,
        IsOwner: &isOwnerMallParaDto,
        Limit: &limitMallParaDto,
        Offset: &offsetMallParaDto,
        Visibility: &visibilityMallParaDto,
    }
}
response, err := client.ListDataServiceMarketApis(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.6 网关管理接口

### 11.6.1 获取网关实例(专享版)

#### 功能介绍

获取网关实例(专享版)。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/service/apigw/instances

表 11-223 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-224 Query 参数

参数	是否必选	参数类型	描述
apig_type	是	String	网关类型。 枚举值: <ul style="list-style-type: none"><li>• APIGW: APIGW类型网关</li><li>• ROMA_APIC: ROMA类型网关</li></ul>
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或LIMIT的整数倍, 不满足则向下取整。

## 请求参数

表 11-225 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID</a> 和 <a href="#">工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-226 响应 Body 参数

参数	参数类型	描述
gateway_instances	Array of <a href="#">ApigInstanceDTO</a> objects	网关实例。

表 11-227 ApigInstanceDTO

参数	参数类型	描述
instance_id	String	网关实例编号。
instance_name	String	网关实例名称。

状态码： 400

表 11-228 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

获取网关实例(专享版)。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apigw/instances
```

## 响应示例

**状态码： 200**

Success。

```
{
  "gateway_instances" : [ {
    "instance_id" : "0ca3311f5dc941b7b3d1950d1682b646",
    "instance_name" : "apigw_test_001"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListApicInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListApicInstancesRequest request = new ListApicInstancesRequest();
try {
    ListApicInstancesResponse response = client.listApicInstances(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListApicInstancesRequest()
        response = client.list_apic_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)
```

```

)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListApicInstancesRequest{}
    response, err := client.ListApicInstances(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.6.2 获取网关分组

### 功能介绍

获取网关分组。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/apigw/instances/{apig\_instance\_id}/groups

表 11-229 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
apig_instance_id	是	String	网关实例编号，共享版为固定值：APIG。

表 11-230 Query 参数

参数	是否必选	参数类型	描述
apig_type	是	String	网关类型。 枚举值： <ul style="list-style-type: none"> <li>• APIG：APIG类型网关</li> <li>• APIGW：APIGW类型网关</li> <li>• ROMA_APIC：ROMA类型网关</li> </ul>
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标，即跳过前X条数据。仅支持0或LIMIT的整数倍，不满足则向下取整。

## 请求参数

表 11-231 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。



参数	是否必选	参数类型	描述
Content-Type	是	String	消息体的类型（格式），有 Body 体的情况下必选，没有 Body 体无需填写。如果请求消息体中含有中文字符，则需要通过 charset=utf8 指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-232 响应 Body 参数

参数	参数类型	描述
gateway_groups	Array of <a href="#">ApigGroupDTO</a> objects	网关分组列表。

表 11-233 ApigGroupDTO

参数	参数类型	描述
group_id	String	分组编号。
group_name	String	分组名称。

状态码： 400

表 11-234 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

获取网关分组。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apigw/instances/51159105c7838353d2834181d978af50/groups
```

## 响应示例

状态码： 200

Success。

```
{
  "gateway_groups": [ {
    "group_id": "ef5a4d1dd3064702a943c2de303348e2",
    "group_name": "DEFAULT"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListApicGroupsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListApicGroupsRequest request = new ListApicGroupsRequest();
        request.withApigInstanceId("{apig_instance_id}");
        try {
            ListApicGroupsResponse response = client.listApicGroups(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListApicGroupsRequest()  
        request.apig_instance_id = "{apig_instance_id}"  
        response = client.list_apic_groups(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListApigGroupsRequest{}
request.ApigInstanceId = "{apig_instance_id}"
response, err := client.ListApigGroups(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

# 11.7 应用管理接口

## 11.7.1 创建应用

### 功能介绍

创建应用。支持创建APP，IAM应用。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/service/apps

表 11-235 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

## 请求参数

表 11-236 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-237 请求 Body 参数

参数	是否必选	参数类型	描述
app_type	否	String	应用类型。 枚举值： <ul style="list-style-type: none"> <li>• APP：APP类型应用</li> <li>• IAM：IAM类型应用</li> <li>• APIG：APIG应用（已废弃）</li> <li>• APIGW：APIGW类型应用（已废弃）</li> <li>• DLM：数据服务应用（已废弃）</li> <li>• ROMA_APIC：ROMA类型应用（已废弃）</li> </ul>
name	否	String	应用名称。
description	否	String	应用描述。

参数	是否必选	参数类型	描述
apig_type	否	String	网关类型。 枚举值： <ul style="list-style-type: none"> <li>• APIG: APIG类型网关</li> <li>• APIGW: APIGW类型网关</li> <li>• ROMA_APIC: ROMA类型网关</li> </ul>
apig_instance_id	否	String	网关实例编号。

## 响应参数

状态码： 200

表 11-238 响应 Body 参数

参数	参数类型	描述
id	String	应用ID编号。
name	String	应用名称。
description	String	应用描述。
app_key	String	应用key。
app_secret	String	应用secret。
register_time	Long	创建时间。
update_time	Long	更新时间。
create_user	String	创建者。
update_user	String	更新者。
app_type	String	应用类型。 枚举值： <ul style="list-style-type: none"> <li>• APP: APP类型应用</li> <li>• IAM: IAM类型应用</li> <li>• APIG: APIG应用（已废弃）</li> <li>• APIGW: APIGW类型应用（已废弃）</li> <li>• DLM: 数据服务应用（已废弃）</li> <li>• ROMA_APIC: ROMA类型应用（已废弃）</li> </ul>

状态码： 400

表 11-239 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

创建应用。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apps
{
  "app_type": "APP",
  "name": "app_test_001",
  "description": "This is the app's description."
}
```

## 响应示例

状态码： 200

Success。

```
{
  "id": "908489209a320df61607355c57c82882",
  "name": "app_test_001",
  "description": "This is the app's description.",
  "app_key": "fbbf9462bb434cb4a13ee7d2bb70c418",
  "app_secret": "c1245f01937442e098ffa6ff134cb3dc",
  "register_time": 1578284788000,
  "update_time": 1578284788000,
  "create_user": "create_user",
  "update_user": "update_user",
  "app_type": "APP"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建应用。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateAppSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    CreateAppRequest request = new CreateAppRequest();
    AppRequestDTO body = new AppRequestDTO();
    body.withDescription("This is the app's description.");
    body.withName("app_test_001");
    body.withAppType(AppRequestDTO.AppTypeEnum.fromValue("APP"));
    request.withBody(body);
    try {
        CreateAppResponse response = client.createApp(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

创建应用。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
```



```
.build()

try:
    request = CreateAppRequest()
    request.body = AppRequestDTO(
        description="This is the app's description.",
        name="app_test_001",
        app_type="APP"
    )
    response = client.create_app(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 创建应用。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAppRequest{}
    descriptionAppRequestDto := "This is the app's description."
    nameAppRequestDto := "app_test_001"
    appTypeAppRequestDto := model.GetAppRequestDtoAppTypeEnum().APP
    request.Body = &model.AppRequestDto{
        Description: &descriptionAppRequestDto,
        Name: &nameAppRequestDto,
        AppType: &appTypeAppRequestDto,
    }
    response, err := client.CreateApp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.7.2 查询应用列表

### 功能介绍

查询应用列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/apps

表 11-240 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-241 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标，即跳过前X条数据。仅支持0或LIMIT的整数倍，不满足则向下取整。
name	否	String	应用名称，该参数为模糊匹配。

参数	是否必选	参数类型	描述
app_type	否	String	应用类型。 枚举值： <ul style="list-style-type: none"> <li>• APP: APP类型应用</li> <li>• IAM: IAM类型应用</li> <li>• APIG: APIG应用（已废弃）</li> <li>• APIGW: APIGW类型应用（已废弃）</li> <li>• DLM: 数据服务应用（已废弃）</li> <li>• ROMA_APIC: ROMA类型应用（已废弃）</li> </ul>

## 请求参数

表 11-242 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-243 响应 Body 参数

参数	参数类型	描述
total	Integer	总数。
apps	Array of <a href="#">AppReturnDTO</a> objects	应用列表。

表 11-244 AppReturnDTO

参数	参数类型	描述
id	String	应用ID编号。
name	String	应用名称。
description	String	应用描述。
app_key	String	应用key。
app_secret	String	应用secret。
register_time	Long	创建时间。
update_time	Long	更新时间。
create_user	String	创建者。
update_user	String	更新者。
app_type	String	应用类型。 枚举值： <ul style="list-style-type: none"> <li>● APP: APP类型应用</li> <li>● IAM: IAM类型应用</li> <li>● APIG: APIG应用（已废弃）</li> <li>● APIGW: APIGW类型应用（已废弃）</li> <li>● DLM: 数据服务应用（已废弃）</li> <li>● ROMA_APIC: ROMA类型应用（已废弃）</li> </ul>

状态码： 400

表 11-245 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询应用列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apps
```

## 响应示例

状态码： 200

Success。

```
{
  "total": 1,
  "apps": [ {
    "id": "908489209a320df61607355c57c82882",
    "name": "app_test_001",
    "description": "This is the app's description.",
    "app_key": "fbbf9462bb434cb4a13ee7d2bb70c418",
    "app_secret": "c1245f01937442e098ffa6ff134cb3dc",
    "register_time": 1578284788000,
    "update_time": 1578284788000,
    "create_user": "create_user",
    "update_user": "update_user",
    "app_type": "APP"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListAppsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
ListAppsRequest request = new ListAppsRequest();
try {
    ListAppsResponse response = client.listApps(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAppsRequest()
        response = client.list_apps(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAppsRequest{}
response, err := client.ListApps(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.7.3 更新应用

### 功能介绍

更新应用。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/service/apps/{app\_id}

表 11-246 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
app_id	是	String	应用ID编号。

## 请求参数

表 11-247 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-248 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	应用名称。
description	否	String	应用描述。

## 响应参数

状态码： 200



表 11-249 响应 Body 参数

参数	参数类型	描述
id	String	应用ID编号。
name	String	应用名称。
description	String	应用描述。
app_key	String	应用key。
app_secret	String	应用secret。
register_time	Long	创建时间。
update_time	Long	更新时间。
create_user	String	创建者。
update_user	String	更新者。
app_type	String	应用类型。 枚举值： <ul style="list-style-type: none"> <li>• APP: APP类型应用</li> <li>• IAM: IAM类型应用</li> <li>• APIG: APIG应用（已废弃）</li> <li>• APIGW: APIGW类型应用（已废弃）</li> <li>• DLM: 数据服务应用（已废弃）</li> <li>• ROMA_APIC: ROMA类型应用（已废弃）</li> </ul>

状态码： 400

表 11-250 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

更新应用。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apps/760715eb1bfce0c575abab3be3bd41e6
{
  "name" : "app_test_001",
  "description" : "This is the app's description."
}
```

## 响应示例

**状态码： 200**

Success。

```
{
  "id" : "908489209a320df61607355c57c82882",
  "name" : "app_test_001",
  "description" : "This is the app's description.",
  "app_key" : "fbbf9462bb434cb4a13ee7d2bb70c418",
  "app_secret" : "c1245f01937442e098ffa6ff134cb3dc",
  "register_time" : 1578284788000,
  "update_time" : 1578284788000,
  "create_user" : "create_user",
  "update_user" : "update_user",
  "app_type" : "APP"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新应用。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateAppSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAppRequest request = new UpdateAppRequest();
        request.withAppId("{app_id}");
        AppUpdateDTO body = new AppUpdateDTO();
        body.withDescription("This is the app's description.");
        body.withName("app_test_001");
        request.withBody(body);
        try {
```

```
        UpdateAppResponse response = client.updateApp(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

更新应用。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateAppRequest()
        request.app_id = "{app_id}"
        request.body = AppUpdateDTO(
            description="This is the app's description.",
            name="app_test_001"
        )
        response = client.update_app(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

更新应用。

```
package main

import (
    "fmt"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAppRequest{}
    request.AppId = "{app_id}"
    descriptionAppUpdateDto:= "This is the app's description."
    nameAppUpdateDto:= "app_test_001"
    request.Body = &model.AppUpdateDto{
        Description: &descriptionAppUpdateDto,
        Name: &nameAppUpdateDto,
    }
    response, err := client.UpdateApp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

## 11.7.4 删除应用

### 功能介绍

删除应用。

## 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v1/{project\_id}/service/apps/{app\_id}

表 11-251 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
app_id	是	String	应用ID编号。

## 请求参数

表 11-252 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 400

表 11-253 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

删除应用。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apps/760715eb1bfce0c575abab3be3bd41e6
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteAppSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAppRequest request = new DeleteAppRequest();
        request.withAppId("{app_id}");
        try {
            DeleteAppResponse response = client.deleteApp(request);
            System.out.println(response.toString());
        }
    }
}
```

```
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteAppRequest()
        request.app_id = "{app_id}"
        response = client.delete_app(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteAppRequest{}
request.AppId = "{app_id}"
response, err := client.DeleteApp(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	Success。
400	BadRequest。

## 11.7.5 查询应用详情

### 功能介绍

查询应用详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/apps/{app\_id}



表 11-254 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
app_id	是	String	应用ID编号。

## 请求参数

表 11-255 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-256 响应 Body 参数

参数	参数类型	描述
id	String	应用ID编号。
name	String	应用名称。
description	String	应用描述。
app_key	String	应用key。
app_secret	String	应用secret。

参数	参数类型	描述
register_time	Long	创建时间。
update_time	Long	更新时间。
create_user	String	创建者。
update_user	String	更新者。
app_type	String	应用类型。 枚举值： <ul style="list-style-type: none"> <li>● APP: APP类型应用</li> <li>● IAM: IAM类型应用</li> <li>● APIG: APIG应用（已废弃）</li> <li>● APIGW: APIGW类型应用（已废弃）</li> <li>● DLM: 数据服务应用（已废弃）</li> <li>● ROMA_APIC: ROMA类型应用（已废弃）</li> </ul>

状态码： 400

表 11-257 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询应用详情。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/apps/760715eb1bfce0c575abab3be3bd41e6
```

## 响应示例

状态码： 200

Success。

```
{
  "id": "908489209a320df61607355c57c82882",
  "name": "app_test_001",
  "description": "This is the app's description.",
  "app_key": "fbbf9462bb434cb4a13ee7d2bb70c418",
  "app_secret": "c1245f01937442e098ffa6ff134cb3dc",
  "register_time": 1578284788000,
  "update_time": 1578284788000,
  "create_user": "create_user",
  "update_user": "update_user",
  "app_type": "APP"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowAppInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAppInfoRequest request = new ShowAppInfoRequest();
        request.withAppId("{app_id}");
        try {
            ShowAppInfoResponse response = client.showAppInfo(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowAppInfoRequest()
        request.app_id = "{app_id}"
        response = client.show_app_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAppInfoRequest{}
    request.AppId = "{app_id}"
    response, err := client.ShowAppInfo(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	Success。
400	BadRequest。

# 11.8 总览接口

## 11.8.1 查询统计用户相关的总览开发指标

### 功能介绍

查询统计用户相关的总览开发指标。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apis-overview

表 11-258 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-259 Query 参数

参数	是否必选	参数类型	描述
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。

参数	是否必选	参数类型	描述
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"> <li>• HOUR：小时为单位</li> <li>• DAY：天为单位</li> </ul>

## 请求参数

表 11-260 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-261 响应 Body 参数

参数	参数类型	描述
publish_num	Integer	已发布API量。
developing_num	Integer	开发中API量。
apply_num	Integer	申请量。
call_num	Integer	调用总量。

参数	参数类型	描述
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。

**状态码： 400**

**表 11-262 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询统计用户相关的总览开发指标。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apis-overview
```

## 响应示例

**状态码： 200**

查询统计用户相关的总览开发指标成功。

```
{
  "publish_num" : 54,
  "developing_num" : 117,
  "apply_num" : 4,
  "call_num" : 10,
  "success_num" : 9,
  "fail_num" : 1,
  "legal_num" : 10,
  "illegal_num" : 0
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowApisOverviewSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowApisOverviewRequest request = new ShowApisOverviewRequest();
        try {
            ShowApisOverviewResponse response = client.showApisOverview(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
```



```

        .build()

    try:
        request = ShowApisOverviewRequest()
        response = client.show_apis_overview(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowApisOverviewRequest{}
    response, err := client.ShowApisOverview(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询统计用户相关的总览开发指标成功。

状态码	描述
400	BadRequest。

## 11.8.2 查询统计用户相关的总览调用指标

### 功能介绍

查询统计用户相关的总览调用指标。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apps-overview

表 11-263 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-264 Query 参数

参数	是否必选	参数类型	描述
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"><li>• HOUR：小时为单位</li><li>• DAY：天为单位</li></ul>

## 请求参数

表 11-265 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必须选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-266 响应 Body 参数

参数	参数类型	描述
apply_num	Integer	申请量。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。

状态码： 400

表 11-267 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询统计用户相关的总览调用指标。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apps-overview
```

## 响应示例

**状态码： 200**

查询统计用户相关的总览调用指标成功。

```
{  
  "apply_num" : 4,  
  "call_num" : 10,  
  "success_num" : 9,  
  "fail_num" : 1,  
  "legal_num" : 10,  
  "illegal_num" : 0  
}
```

## 状态码

状态码	描述
200	查询统计用户相关的总览调用指标成功。
400	BadRequest。

## 11.8.3 查询 API 服务调用 TopN

### 功能介绍

查询API服务调用TopN。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apis-top-n

表 11-268 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-269 Query 参数

参数	是否必选	参数类型	描述
instance_id	否	String	集群ID编号。
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"> <li>• HOUR：小时为单位</li> <li>• DAY：天为单位</li> </ul>
top_num	是	Integer	前几名。
order_by	是	String	排序条件。 枚举值： <ul style="list-style-type: none"> <li>• CALL_NUM：总调用次数</li> <li>• SUCCESS_NUM：调用成功次数</li> <li>• FAIL_NUM：调用失败次数</li> <li>• LEGAL_NUM：合法次数</li> <li>• ILLEGAL_NUM：非法次数</li> <li>• COST_TIME_AVG：平均时长</li> <li>• SUCCESS_COST_TIME_AVG：成功平均时长</li> <li>• FAIL_COST_TIME_AVG：失败平均时长</li> <li>• SUCCESS_RATE：成功率</li> <li>• FAIL_RATE：失败率</li> <li>• LEGAL_RATE：合法率</li> <li>• ILLEGAL_RATE：非法率</li> </ul>

## 请求参数

表 11-270 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-271 响应 Body 参数

参数	参数类型	描述
statistics	Array of <a href="#">StatisticForDetail</a> objects	统计信息详情列表。

表 11-272 StatisticForDetail

参数	参数类型	描述
id	String	统计对象编号。
name	String	统计对象名称。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。

参数	参数类型	描述
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。
cost_time_avg	Number	请求平均时长。
success_cost_time_avg	Number	成功请求平均时长。
fail_cost_time_avg	Number	失败请求平均时长。
success_rate	Number	成功率。
fail_rate	Number	失败率。
legal_rate	Number	合法率。
illegal_rate	Number	非法率。

**状态码： 400**

**表 11-273 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询API服务调用TopN。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apis-top-n
```

## 响应示例

**状态码： 200**

查询API服务调用TopN成功。

```
{
  "statistics": [ {
    "id": "3423634676234236674",
    "name": "统计对象名称",
    "call_num": 0,
    "success_num": 0,
    "fail_num": 0,
    "legal_num": 0,
    "illegal_num": 0,
    "cost_time_avg": 0,
    "success_cost_time_avg": 0,
    "fail_cost_time_avg": 0,
  }
]
```

```
"success_rate" : 0,  
"fail_rate" : 0,  
"legal_rate" : 0,  
"illegal_rate" : 0  
} ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListApisTopSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListApisTopRequest request = new ListApisTopRequest();  
        try {  
            ListApisTopResponse response = client.listApisTop(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8
```



```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListApisTopRequest()
        response = client.list_apis_top(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListApisTopRequest{}
    response, err := client.ListApisTop(request)
```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询API服务调用TopN成功。
400	BadRequest。

## 11.8.4 查询 APP 服务使用 TopN

### 功能介绍

查询APP服务使用TopN。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apps-top-n

表 11-274 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-275 Query 参数

参数	是否必选	参数类型	描述
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。

参数	是否必选	参数类型	描述
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"> <li>• HOUR：小时为单位</li> <li>• DAY：天为单位</li> </ul>
top_num	是	Integer	前几名。
order_by	是	String	排序条件。 枚举值： <ul style="list-style-type: none"> <li>• CALL_NUM：总调用次数</li> <li>• SUCCESS_NUM：调用成功次数</li> <li>• FAIL_NUM：调用失败次数</li> <li>• LEGAL_NUM：合法次数</li> <li>• ILLEGAL_NUM：非法次数</li> <li>• COST_TIME_AVG：平均时长</li> <li>• SUCCESS_COST_TIME_AVG：成功平均时长</li> <li>• FAIL_COST_TIME_AVG：失败平均时长</li> <li>• SUCCESS_RATE：成功率</li> <li>• FAIL_RATE：失败率</li> <li>• LEGAL_RATE：合法率</li> <li>• ILLEGAL_RATE：非法率</li> </ul>

## 请求参数

表 11-276 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。

参数	是否必选	参数类型	描述
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-277 响应 Body 参数

参数	参数类型	描述
statistics	Array of <a href="#">StatisticForDetail</a> objects	统计信息详情列表。

表 11-278 StatisticForDetail

参数	参数类型	描述
id	String	统计对象编号。
name	String	统计对象名称。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。
cost_time_avg	Number	请求平均时长。
success_cost_time_avg	Number	成功请求平均时长。
fail_cost_time_avg	Number	失败请求平均时长。
success_rate	Number	成功率。
fail_rate	Number	失败率。
legal_rate	Number	合法率。

参数	参数类型	描述
illegal_rate	Number	非法率。

状态码： 400

表 11-279 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询APP服务使用TopN。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apps-top-n
```

## 响应示例

状态码： 200

查询APP服务使用TopN成功。

```
{
  "statistics": [ {
    "id": "3423634676234236674",
    "name": "统计对象名称",
    "call_num": 0,
    "success_num": 0,
    "fail_num": 0,
    "legal_num": 0,
    "illegal_num": 0,
    "cost_time_avg": 0,
    "success_cost_time_avg": 0,
    "fail_cost_time_avg": 0,
    "success_rate": 0,
    "fail_rate": 0,
    "legal_rate": 0,
    "illegal_rate": 0
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListAppsTopSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAppsTopRequest request = new ListAppsTopRequest();
        try {
            ListAppsTopResponse response = client.listAppsTop(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = ListAppsTopRequest()  
    response = client.list_apps_top(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dataartsstudio.NewDataArtsStudioClient(  
        dataartsstudio.DataArtsStudioClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListAppsTopRequest{}  
    response, err := client.ListAppsTop(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询APP服务使用TopN成功。
400	BadRequest。

## 11.8.5 查询 API 统计数据详情

### 功能介绍

查询API统计数据详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apis-detail/{api\_id}

表 11-280 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。

表 11-281 Query 参数

参数	是否必选	参数类型	描述
instance_id	否	String	集群ID编号。
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"><li>• HOUR：小时为单位</li><li>• DAY：天为单位</li></ul>



## 请求参数

表 11-282 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-283 响应 Body 参数

参数	参数类型	描述
id	String	统计对象编号。
name	String	统计对象名称。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。
cost_time_avg	Number	请求平均时长。
success_cost_time_avg	Number	成功请求平均时长。
fail_cost_time_avg	Number	失败请求平均时长。

参数	参数类型	描述
success_rate	Number	成功率。
fail_rate	Number	失败率。
legal_rate	Number	合法率。
illegal_rate	Number	非法率。

状态码： 400

表 11-284 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询指定API统计数据详情。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apis-detail/  
760715eb1bfce0c575abab3be3bd41e6
```

## 响应示例

状态码： 200

查询API统计数据详情成功。

```
{  
  "id" : "3423634676234236674",  
  "name" : "统计对象名称",  
  "call_num" : 0,  
  "success_num" : 0,  
  "fail_num" : 0,  
  "legal_num" : 0,  
  "illegal_num" : 0,  
  "cost_time_avg" : 0,  
  "success_cost_time_avg" : 0,  
  "fail_cost_time_avg" : 0,  
  "success_rate" : 0,  
  "fail_rate" : 0,  
  "legal_rate" : 0,  
  "illegal_rate" : 0  
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowApisDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowApisDetailRequest request = new ShowApisDetailRequest();
        request.withApiId("{api_id}");
        try {
            ShowApisDetailResponse response = client.showApisDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowApisDetailRequest()
    request.api_id = "{api_id}"
    response = client.show_apis_detail(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowApisDetailRequest{}
    request.ApiId = "{api_id}"
    response, err := client.ShowApisDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询API统计数据详情成功。
400	BadRequest。

## 11.8.6 查询 APP 统计数据详情

### 功能介绍

查询APP统计数据详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apps-detail/{app\_id}

表 11-285 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
app_id	是	String	APP编号。

表 11-286 Query 参数

参数	是否必选	参数类型	描述
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"> <li>• HOUR：小时为单位</li> <li>• DAY：天为单位</li> </ul>

## 请求参数

表 11-287 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。APP编号
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-288 响应 Body 参数

参数	参数类型	描述
id	String	统计对象编号。
name	String	统计对象名称。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。
cost_time_avg	Number	请求平均时长。
success_cost_time_avg	Number	成功请求平均时长。
fail_cost_time_avg	Number	失败请求平均时长。

参数	参数类型	描述
success_rate	Number	成功率。
fail_rate	Number	失败率。
legal_rate	Number	合法率。
illegal_rate	Number	非法率。

状态码： 400

表 11-289 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询指定APP统计数据详情。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apps-detail/760715eb1bfce0c575abab3be3bd41e6
```

## 响应示例

状态码： 200

查询APP统计数据详情成功。

```
{
  "id": "3423634676234236674",
  "name": "统计对象名称",
  "call_num": 0,
  "success_num": 0,
  "fail_num": 0,
  "legal_num": 0,
  "illegal_num": 0,
  "cost_time_avg": 0,
  "success_cost_time_avg": 0,
  "fail_cost_time_avg": 0,
  "success_rate": 0,
  "fail_rate": 0,
  "legal_rate": 0,
  "illegal_rate": 0
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowAppsDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAppsDetailRequest request = new ShowAppsDetailRequest();
        request.withAppId("{app_id}");
        try {
            ShowAppsDetailResponse response = client.showAppsDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```



example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowAppsDetailRequest()
    request.app_id = "{app_id}"
    response = client.show_apps_detail(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAppsDetailRequest{}
    request.AppId = "{app_id}"
    response, err := client.ShowAppsDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询APP统计数据详情成功。
400	BadRequest。

## 11.8.7 查询 API 仪表盘数据详情

### 功能介绍

查询API仪表盘数据详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apis-dashboards

表 11-290 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-291 Query 参数

参数	是否必选	参数类型	描述
instance_id	否	String	集群ID编号。
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"> <li>• HOUR：小时为单位</li> <li>• DAY：天为单位</li> </ul>

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标, 即跳过前X条数据。仅支持0或LIMIT的整数倍, 不满足则向下取整。

## 请求参数

表 11-292 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token, 使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型, 指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型(格式), 有Body体的情况下必选, 没有Body体无需填写。如果请求消息体中含有中文字符, 则需要通过charset=utf8指定中文字符集, 例如取值为: application/json;charset=utf8。

## 响应参数

状态码: 200

表 11-293 响应 Body 参数

参数	参数类型	描述
dashboards	Array of <a href="#">StatisticForDashboard</a> objects	统计信息仪表盘。

表 11-294 StatisticForDashboard

参数	参数类型	描述
log_time	Integer	统计时间点(13位时间戳)。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。
cost_time_avg	Number	请求平均时长。
success_cost_time_avg	Number	成功请求平均时长。
fail_cost_time_avg	Number	失败请求平均时长。
success_rate	Number	成功率。
fail_rate	Number	失败率。
legal_rate	Number	合法率。
illegal_rate	Number	非法率。

状态码： 400

表 11-295 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询API仪表板数据详情。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apis-dashboards
```

## 响应示例

状态码： 200

查询API仪表板数据详情成功。

```
{
  "dashboards": [ {
```

```
"log_time" : 1629403200000,
"call_num" : 0,
"success_num" : 0,
"fail_num" : 0,
"legal_num" : 0,
"illegal_num" : 0,
"cost_time_avg" : 0,
"success_cost_time_avg" : 0,
"fail_cost_time_avg" : 0,
"success_rate" : 0,
"fail_rate" : 0,
"legal_rate" : 0,
"illegal_rate" : 0
} ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowApisDashboardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowApisDashboardRequest request = new ShowApisDashboardRequest();
        try {
            ShowApisDashboardResponse response = client.showApisDashboard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowApisDashboardRequest()
        response = client.show_apis_dashboard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowApisDashboardRequest{}
response, err := client.ShowApisDashboard(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询API仪表盘数据详情成功。
400	BadRequest。

## 11.8.8 查询指定 API 仪表盘数据详情

### 功能介绍

查询指定API仪表盘数据详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apis-dashboards/{api\_id}

表 11-296 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。

表 11-297 Query 参数

参数	是否必选	参数类型	描述
instance_id	否	String	集群ID编号。
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"> <li>• HOUR：小时为单位</li> <li>• DAY：天为单位</li> </ul>

## 请求参数

表 11-298 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <b>Token认证</b> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200



表 11-299 响应 Body 参数

参数	参数类型	描述
dashboards	Array of <a href="#">StatisticForDashboard</a> objects	统计信息仪表板。

表 11-300 StatisticForDashboard

参数	参数类型	描述
log_time	Integer	统计时间点(13位时间戳)。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。
cost_time_avg	Number	请求平均时长。
success_cost_time_avg	Number	成功请求平均时长。
fail_cost_time_avg	Number	失败请求平均时长。
success_rate	Number	成功率。
fail_rate	Number	失败率。
legal_rate	Number	合法率。
illegal_rate	Number	非法率。

状态码： 400

表 11-301 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询指定API仪表盘数据详情。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apis-dashboards/  
760715eb1bfce0c575abab3be3bd41e6
```

## 响应示例

**状态码： 200**

查询指定APP仪表盘数据详情成功。

```
{  
  "dashboards": [ {  
    "log_time": 1629403200000,  
    "call_num": 0,  
    "success_num": 0,  
    "fail_num": 0,  
    "legal_num": 0,  
    "illegal_num": 0,  
    "cost_time_avg": 0,  
    "success_cost_time_avg": 0,  
    "fail_cost_time_avg": 0,  
    "success_rate": 0,  
    "fail_rate": 0,  
    "legal_rate": 0,  
    "illegal_rate": 0  
  } ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowApiDashboardSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowApiDashboardRequest request = new ShowApiDashboardRequest();
request.withApild("{api_id}");
try {
    ShowApiDashboardResponse response = client.showApiDashboard(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowApiDashboardRequest()
        request.api_id = "{api_id}"
        response = client.show_api_dashboard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowApiDashboardRequest{}
    request.ApiId = "{api_id}"
    response, err := client.ShowApiDashboard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询指定APP仪表板数据详情成功。
400	BadRequest。

## 11.8.9 查询 APP 仪表板数据详情

### 功能介绍

查询APP仪表板数据详情。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/service/statistic/apps-dashboards

表 11-302 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-303 Query 参数

参数	是否必选	参数类型	描述
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"> <li>• HOUR：小时为单位</li> <li>• DAY：天为单位</li> </ul>
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标，即跳过前X条数据。仅支持0或LIMIT的整数倍，不满足则向下取整。

## 请求参数

表 11-304 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <a href="#">Token认证</a> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。

参数	是否必选	参数类型	描述
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-305 响应 Body 参数

参数	参数类型	描述
dashboards	Array of <b>StatisticForDashboard</b> objects	统计信息仪表盘。

表 11-306 StatisticForDashboard

参数	参数类型	描述
log_time	Integer	统计时间点(13位时间戳)。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。
cost_time_avg	Number	请求平均时长。
success_cost_time_avg	Number	成功请求平均时长。
fail_cost_time_avg	Number	失败请求平均时长。
success_rate	Number	成功率。
fail_rate	Number	失败率。
legal_rate	Number	合法率。

参数	参数类型	描述
illegal_rate	Number	非法率。

状态码： 400

表 11-307 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询APP仪表板数据详情。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apps-dashboards
```

## 响应示例

状态码： 200

查询APP仪表板数据详情成功。

```
{
  "dashboards": [ {
    "log_time": 1629403200000,
    "call_num": 0,
    "success_num": 0,
    "fail_num": 0,
    "legal_num": 0,
    "illegal_num": 0,
    "cost_time_avg": 0,
    "success_cost_time_avg": 0,
    "fail_cost_time_avg": 0,
    "success_rate": 0,
    "fail_rate": 0,
    "legal_rate": 0,
    "illegal_rate": 0
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowAppsDashboardSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAppsDashboardRequest request = new ShowAppsDashboardRequest();
        try {
            ShowAppsDashboardResponse response = client.showAppsDashboard(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
```



```

        .build()

    try:
        request = ShowAppsDashboardRequest()
        response = client.show_apps_dashboard(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAppsDashboardRequest{}
    response, err := client.ShowAppsDashboard(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询APP仪表盘数据详情成功。

状态码	描述
400	BadRequest。

## 11.8.10 查询指定 API 应用调用 TopN

### 功能介绍

查询指定API应用调用TopN。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/statistic/apis-top-n/{api\_id}

表 11-308 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
api_id	是	String	API编号。

表 11-309 Query 参数

参数	是否必选	参数类型	描述
instance_id	否	String	集群ID编号。
start_time	是	Long	开始时间（13位时间戳）。
end_time	是	Long	结束时间（13位时间戳）。
time_unit	是	String	时间单位。 枚举值： <ul style="list-style-type: none"> <li>• HOUR：小时为单位</li> <li>• DAY：天为单位</li> </ul>
top_num	是	Integer	前几名。

参数	是否必选	参数类型	描述
order_by	是	String	<p>排序条件。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• CALL_NUM: 总调用次数</li> <li>• SUCCESS_NUM: 调用成功次数</li> <li>• FAIL_NUM: 调用失败次数</li> <li>• LEGAL_NUM: 合法次数</li> <li>• ILLEGAL_NUM: 非法次数</li> <li>• COST_TIME_AVG: 平均时长</li> <li>• SUCCESS_COST_TIME_AVG : 成功平均时长</li> <li>• FAIL_COST_TIME_AVG: 失败平均时长</li> <li>• SUCCESS_RATE: 成功率</li> <li>• FAIL_RATE: 失败率</li> <li>• LEGAL_RATE: 合法率</li> <li>• ILLEGAL_RATE: 非法率</li> </ul>

## 请求参数

表 11-310 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用 <b>Token认证</b> 时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	是	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-311 响应 Body 参数

参数	参数类型	描述
statistics	Array of <b>StatisticForCallDetail</b> objects	调用信息列表。

表 11-312 StatisticForCallDetail

参数	参数类型	描述
id	String	统计对象编号。
name	String	统计对象名称。
call_num	Integer	调用总量。
success_num	Integer	成功调用量(取数成功)。
fail_num	Integer	失败调用量(取数失败)。
legal_num	Integer	合法调用量(通过校验)。
illegal_num	Integer	非法调用量(无法通过校验)。

状态码： 400

表 11-313 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询指定API应用调用TopN。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/statistic/apis-top-n/760715eb1bfce0c575abab3be3bd41e6
```

## 响应示例

状态码： 200

查询指定api 应用调用TopN成功。

```
{
  "statistics": [ {
    "id": "3423634676234236674",
    "name": "统计对象名称",
    "call_num": 0,
    "success_num": 0,
    "fail_num": 0,
    "legal_num": 0,
    "illegal_num": 0
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListApiTopNSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListApiTopNRequest request = new ListApiTopNRequest();
        request.withApiId("{api_id}");
        try {
            ListApiTopNResponse response = client.listApiTopN(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListApiTopNRequest()  
        request.api_id = "{api_id}"  
        response = client.list_api_top_n(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListApiTopNRequest{}
request.ApiId = "{api_id}"
response, err := client.ListApiTopN(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询指定api 应用调用TopN成功。
400	BadRequest。

# 11.9 集群管理接口

## 11.9.1 查询集群概览信息列表

### 功能介绍

查询集群概览信息列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/instances/overview

表 11-314 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-315 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标。
name	否	String	集群名称，该参数为模糊匹配。
create_user	否	String	创建人名称。

## 请求参数

表 11-316 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	否	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-317 响应 Body 参数

参数	参数类型	描述
total	Integer	集群数量。
scale_down	Boolean	是否支持缩容。
scale_out	Boolean	是否支持扩容。



参数	参数类型	描述
instances	Array of <b>InstanceOverviewDTO</b> objects	集群概览信息。

表 11-318 InstanceOverviewDTO

参数	参数类型	描述
id	String	集群ID。
name	String	集群名称。
description	String	集群描述信息。
external_address	String	公网IP地址。
intranet_address	String	内网IPv4地址。
intranet_address_ipv6	String	内网IPv6地址。
public_zone_id	String	公网域名ID。
public_zone_name	String	公网域名名称。
private_zone_id	String	内网域名ID。
private_zone_name	String	内网域名名称。
enterprise_project_id	String	企业项目ID。
create_time	Long	创建时间。
create_user	String	创建人。
current_namespace_published_api_num	Integer	当前工作空间已发布的API数量。
all_namespace_published_api_num	Integer	所有工作空间已发布的API数量。
api_publishable_num	Integer	集群API总配额。

参数	参数类型	描述
deletable	Boolean	集群是否可以删除。
charge_status	String	集群计费状态，NO_CHARGE：未计费、CHARGED：已计费，GRACE：宽限期、RETENTION：保留期。
order_id	String	订单ID。

状态码： 400

表 11-319 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询集群概览信息列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/instances/overview
```

## 响应示例

状态码： 200

查询成功。

```
{
  "total": 1,
  "scale_down": false,
  "scale_out": false,
  "instances": [ {
    "id": "c830776ec23fdffe237a44845f871180",
    "name": "d1m_HCS_20240205",
    "description": "",
    "external_address": "100.85.122.140",
    "intranet_address": "192.168.0.157",
    "intranet_address_ipv6": "2407:c080:11f0:535:8b8f:1825:a0d8:669",
    "private_zone_id": null,
    "private_zone_name": null,
    "public_zone_id": null,
    "public_zone_name": null,
    "enterprise_project_id": "0",
    "create_time": 1708400912000,
    "create_user": "user",
    "current_namespace_publish_api_num": 1,
    "all_namespace_publish_api_num": 1,
    "api_publishable_num": 500,
    "deletable": true,
    "charge_status": "CHARGED",
    "order_id": "c0c3fd9770a54baa8a430ef82218732f"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDataServiceInstancesOverviewSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDataServiceInstancesOverviewRequest request = new ListDataServiceInstancesOverviewRequest();
        try {
            ListDataServiceInstancesOverviewResponse response =
client.listDataServiceInstancesOverview(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListDataServiceInstancesOverviewRequest()
        response = client.list_data_service_instances_overview(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDataServiceInstancesOverviewRequest{}
    response, err := client.ListDataServiceInstancesOverview(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询成功。
400	BadRequest。

## 11.9.2 查询集群详情信息列表

### 功能介绍

查询集群详情信息列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/instances

表 11-320 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 11-321 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数限制。
offset	否	Integer	查询起始坐标。
name	否	String	集群名称，该参数为模糊匹配。
create_user	否	String	创建人名称。

## 请求参数

表 11-322 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	否	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-323 响应 Body 参数

参数	参数类型	描述
total	Integer	集群数量。
instances	Array of InstanceDetailDTO objects	集群概览信息。

表 11-324 InstanceDetailDTO

参数	参数类型	描述
id	String	集群ID。
name	String	集群名称。
description	String	集群描述信息。

参数	参数类型	描述
external_address	String	公网IP地址。
intranet_address	String	内网IPv4地址。
intranet_address_ipv6	String	内网IPv6地址。
public_zone_id	String	公网域名ID。
public_zone_name	String	公网域名名称。
private_zone_id	String	内网域名ID。
private_zone_name	String	内网域名名称。
enterprise_project_id	String	企业项目ID。
create_time	Long	创建时间。
create_user	String	创建人。
current_namespace_publish_api_num	Integer	当前工作空间已发布的API数量。
all_namespace_publish_api_num	Integer	所有工作空间已发布的API数量。
api_publishable_num	Integer	集群API总配额。
deletable	Boolean	集群是否可以删除。
charge_status	String	集群计费状态，NO_CHARGE：未计费、CHARGED：已计费，GRACE：宽限期、RETENTION：保留期。
order_id	String	订单ID。
order_type	String	订单类型，PERIOD：包周期、ON_DEMAND：按需。
period_type	String	集群订购周期类型。
instance_statuses	String	集群状态。
node_num	Integer	节点数量。

参数	参数类型	描述
flavor	<b>FlavorDTO</b> object	集群节点规格。
gateway_version	String	集群版本号。
availability_zone	String	集群所在可用区编码。
availability_zone_name	String	集群所在可用区名称。
vpc_id	String	集群所在虚拟私有云ID。
subnet_id	String	集群所在子网ID。
security_group_id	String	集群所在安全组ID。
nodes	Array of <b>InstanceNodeDTO</b> objects	集群节点列表。

表 11-325 FlavorDTO

参数	参数类型	描述
id	String	规格ID。
name	String	规格名称。
disk	Integer	磁盘大小。
cpu	Integer	CPU大小。
mem	Integer	内存大小。

表 11-326 InstanceNodeDTO

参数	参数类型	描述
id	String	节点ID。
name	String	节点名称。
private_ip	String	节点IP地址。
status	String	节点状态。
create_user	String	节点创建人。
create_time	String	节点创建时间。



参数	参数类型	描述
gateway_version	String	节点版本号。

状态码： 400

表 11-327 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询集群详情信息列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/instances
```

## 响应示例

状态码： 200

查询成功。

```
{
  "total": 1,
  "instances": [ {
    "node_num": 1,
    "flavor": {
      "id": "95d4e992-6e3c-11ea-8e90-fa163e35",
      "name": "测试专用小规格(X86)",
      "disk": 120,
      "cpu": 8,
      "mem": 16
    },
    "gateway_version": "3.0.5",
    "availability_zone": "cn-north-7c",
    "availability_zone_name": "可用区3",
    "vpc_id": "791a1ee9-aacf-477a-b00e-dfb18aa364eb",
    "subnet_id": "43de865d-e4ce-48c3-8c92-f4972026f4c4",
    "security_group_id": "e7527b7f-03d2-49b8-9e78-f4fc3886e792",
    "nodes": [ {
      "id": "49fdf486-af68-4967-8d96-2aca93dc2dbb",
      "name": "dml_HCS_20240205-dayu_dlm-instance-1-1",
      "private_ip": "192.168.0.130",
      "status": "RUNNING",
      "create_user": "user",
      "create_time": "2024-02-20T03:59:25Z",
      "gateway_version": "3.0.5"
    } ],
    "id": "c830776ec23fdffe237a44845f871180",
    "name": "dml_HCS_20240205",
    "description": "",
    "external_address": "100.85.122.140",
    "intranet_address": "192.168.0.157",
    "intranet_address_ipv6": "2407:c080:11f0:535:8b8f:1825:a0d8:669",
  } ]
}
```

```
"private_zone_id" : null,
"private_zone_name" : null,
"public_zone_id" : null,
"public_zone_name" : null,
"instance_status" : "RUNNING",
"enterprise_project_id" : "0",
"create_time" : 1708400912000,
"create_user" : "user",
"current_namespace_publish_api_num" : null,
"all_namespace_publish_api_num" : null,
"api_publishable_num" : null,
"deletable" : true,
"charge_status" : "CHARGED",
"period_type" : "ABSOLUTE",
"order_type" : "ON_DEMAND",
"order_id" : null
}]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListDataServiceInstancesDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDataServiceInstancesDetailRequest request = new ListDataServiceInstancesDetailRequest();
        try {
            ListDataServiceInstancesDetailResponse response = client.listDataServiceInstancesDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

if `__name__ == "__main__"`:

# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables `CLOUD_SDK_AK` and `CLOUD_SDK_SK` in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListDataServiceInstancesDetailRequest()
    response = client.list_data_service_instances_detail(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```

```

WithSk(sk).
WithProjectId(projectId).
Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListDataServiceInstancesDetailRequest{}
response, err := client.ListDataServiceInstancesDetail(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询成功。
400	BadRequest。

## 11.9.3 查询集群详情信息

### 功能介绍

查询集群详情信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/instances/{instance\_id}

表 11-328 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	集群ID。

## 请求参数

表 11-329 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	否	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-330 响应 Body 参数

参数	参数类型	描述
id	String	集群ID。
name	String	集群名称。
description	String	集群描述信息。
external_address	String	公网IP地址。
intranet_address	String	内网IPv4地址。
intranet_address_ipv6	String	内网IPv6地址。
public_zone_id	String	公网域名ID。
public_zone_name	String	公网域名名称。

参数	参数类型	描述
private_zone_id	String	内网域名ID。
private_zone_name	String	内网域名名称。
enterprise_project_id	String	企业项目ID。
create_time	Long	创建时间。
create_user	String	创建人。
current_namespace_publish_api_num	Integer	当前工作空间已发布的API数量。
all_namespace_publish_api_num	Integer	所有工作空间已发布的API数量。
api_publishable_num	Integer	集群API总配额。
deletable	Boolean	集群是否可以删除。
charge_status	String	集群计费状态，NO_CHARGE：未计费、CHARGED：已计费，GRACE：宽限期、RETENTION：保留期。
order_id	String	订单ID。
order_type	String	订单类型，PERIOD：包周期、ON_DEMAND：按需。
period_type	String	集群订购周期类型。
instance_statuses	String	集群状态。
node_num	Integer	节点数量。
flavor	FlavorDTO object	集群节点规格。
gateway_version	String	集群版本号。
availability_zone	String	集群所在可用区编码。
availability_zone_name	String	集群所在可用区名称。
vpc_id	String	集群所在虚拟私有云ID。
subnet_id	String	集群所在子网ID。

参数	参数类型	描述
security_group_id	String	集群所在安全组ID。
nodes	Array of InstanceNodeDTO objects	集群节点列表。

表 11-331 FlavorDTO

参数	参数类型	描述
id	String	规格ID。
name	String	规格名称。
disk	Integer	磁盘大小。
cpu	Integer	CPU大小。
mem	Integer	内存大小。

表 11-332 InstanceNodeDTO

参数	参数类型	描述
id	String	节点ID。
name	String	节点名称。
private_ip	String	节点IP地址。
status	String	节点状态。
create_user	String	节点创建人。
create_time	String	节点创建时间。
gateway_version	String	节点版本号。

状态码： 400

表 11-333 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询id为51159105c7838353d2834181d978af50的集群详情信息。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/instances/51159105c7838353d2834181d978af50
```

## 响应示例

**状态码： 200**

查询成功。

```
{
  "node_num" : 1,
  "flavor" : {
    "id" : "95d4e992-6e3c-11ea-8e90-fa163e35",
    "name" : "测试专用小规格(X86)",
    "disk" : 120,
    "cpu" : 8,
    "mem" : 16
  },
  "gateway_version" : "3.0.5",
  "availability_zone" : "cn-north-7c",
  "availability_zone_name" : "可用区3",
  "vpc_id" : "791a1ee9-aacf-477a-b00e-dfb18aa364eb",
  "subnet_id" : "43de865d-e4ce-48c3-8c92-f4972026f4c4",
  "security_group_id" : "e7527b7f-03d2-49b8-9e78-f4fc3886e792",
  "nodes" : [ {
    "id" : "49fdf486-af68-4967-8d96-2aca93dc2dbb",
    "name" : "d1m_HCS_20240205-dayu_d1m-instance-1-1",
    "private_ip" : "192.168.0.130",
    "status" : "RUNNING",
    "create_user" : "user",
    "create_time" : "2024-02-20T03:59:25Z",
    "gateway_version" : "3.0.5"
  } ],
  "id" : "c830776ec23fdffe237a44845f871180",
  "name" : "d1m_HCS_20240205",
  "description" : "",
  "external_address" : "100.85.122.140",
  "intranet_address" : "192.168.0.157",
  "intranet_address_ipv6" : "2407:c080:11f0:535:8b8f:1825:a0d8:669",
  "private_zone_id" : null,
  "private_zone_name" : null,
  "public_zone_id" : null,
  "public_zone_name" : null,
  "instance_status" : "RUNNING",
  "enterprise_project_id" : "0",
  "create_time" : 1708400912000,
  "create_user" : "user",
  "current_namespace_publish_api_num" : null,
  "all_namespace_publish_api_num" : null,
  "api_publishable_num" : null,
  "deletable" : true,
  "charge_status" : "CHARGED",
  "period_type" : "ABSOLUTE",
  "order_type" : "ON_DEMAND",
  "order_id" : null
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;
```



```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowDataServiceInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDataServiceInstanceRequest request = new ShowDataServiceInstanceRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowDataServiceInstanceResponse response = client.showDataServiceInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowDataServiceInstanceRequest()
    request.instance_id = "{instance_id}"
    response = client.show_data_service_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDataServiceInstanceRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowDataServiceInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	查询成功。
400	BadRequest。

## 11.9.4 查询数据服务集群访问日志列表

### 功能介绍

查询数据服务集群访问日志列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/service/instances/{instance\_id}/accesslog

表 11-334 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	集群ID编号。

表 11-335 Query 参数

参数	是否必选	参数类型	描述
is_api	否	Boolean	是否查询API的访问日志，true 表示查询API的访问日志，false 表示查询应用的访问日志。

## 请求参数

表 11-336 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Content-Type	否	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

## 响应参数

状态码： 200

表 11-337 响应 Body 参数

参数	参数类型	描述
number	Integer	访问日志数量。
records	Array of InstanceAccesslog objects	访问日志列表。

表 11-338 InstanceAccesslog

参数	参数类型	描述
id	String	集群ID。
request_id	String	请求ID。
api_id	String	API ID。
api_name	String	API名称。
app_id	String	APP ID。
app_name	String	APP名称。

参数	参数类型	描述
access_time	Long	访问时间。
duration	Long	访问时长。
status_code	String	状态码。
error_code	String	错误码。
error_message	String	错误信息。
detail	String	日志详情。
in_flow_size	Long	输入流量大小。
out_flow_size	Long	输出流量大小。
out_total_size	Long	输出数据条数。

状态码： 400

表 11-339 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

查询id为4f9366ce3e87860934c36d0ac027ad6a的集群访问日志列表。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/instances/4f9366ce3e87860934c36d0ac027ad6a/accesslog
```

## 响应示例

状态码： 200

查询成功。

```
{
  "records": [ {
    "id": "00006e83b668b67defff1259962909740634112",
    "request_id": "1259962909740634112",
    "api_id": "71a96065c1e0bf11d0914c0658951246",
    "api_name": "iam_post_query",
    "app_id": null,
    "app_name": null,
    "access_time": 1720440166661,
    "duration": 33,
    "status_code": "200",
    "error_code": null,
    "error_message": null,
  }
]
```

```

"detail" : "_____Verification information_____ \nAPI authentication verification: completed.\nApp
authentication verification: completed.\nAccess permission authentication: completed.\n
\n_____Parameter information_____ \n[path parameter]: \n\n[header parameter]: \n\n[nx-language :
en-us\naccept : */*\n\n[query parameter]: \nna : 1\n\n[body parameter]: \n\n_____SQL
information_____ \nSELECT \"bt_col1\" AS \"bt_col1\", \"bt_col2\" AS \"bt_col2\", \"coll\" AS \"coll
\", \"bool\" AS \"bool\", \"dfgh\" AS \"dfgh\", \"dai\" AS \"dai\", \"b1\" AS \"b1\", \"b2\" AS \"b2\", \"ct_col1\"
AS \"ct_col1\", \"id\" AS \"id\", \"col2\" AS \"col2\", \"col1\" AS \"col1\", \"ct_col6\" AS \"ct_col6\", \"a\" AS \"a
\", \"ct_col7\" AS \"ct_col7\", \"b\" AS \"b\", \"ct_col8\" AS \"ct_col8\", \"c\" AS \"c\", \"ct_col9\" AS
\"ct_col9\", \"set\" AS \"set\", \"ct_col2\" AS \"ct_col2\", \"d\" AS \"d\", \"ct_col3\" AS \"ct_col3\", \"ct_col4\" AS
\"ct_col4\", \"ct_col5\" AS \"ct_col5\", \"m\" AS \"m\", \"a1\" AS \"a1\", \"dfga\" AS \"dfga\", \"bt_col31\" AS
\"bt_col31\", \"bt_col11\" AS \"bt_col11\", \"vbg\" AS \"vbg\", \"da\" AS \"da\", \"bt_col3\" AS
\"bt_col3\", \"bt_col4\" AS \"bt_col4\" FROM \"public\".\"dws_all_char_type\" WHERE \"a\" = 1 limit 100
offset 0\n\n",
  "in_flow_size" : 14051,
  "out_flow_size" : 416,
  "out_total_size" : 5
},
"number" : 1
}
    
```

## 状态码

状态码	描述
200	查询成功。
400	BadRequest。

## 11.9.5 开启数据服务集群 OBS 日志转储

### 功能介绍

开启数据服务集群OBS日志转储。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/service/instances/{instance\_id}/obs-log-dump

表 11-340 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	集群ID编号。

## 请求参数

表 11-341 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	否	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-342 请求 Body 参数

参数	是否必选	参数类型	描述
log_dump	否	Boolean	是否开启Obs日志转储功能，true表示开启，false表示关闭。

## 响应参数

状态码： 400

表 11-343 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

开启id为4f9366ce3e87860934c36d0ac027ad6a的集群OBS日志转储功能。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/instances/4f9366ce3e87860934c36d0ac027ad6a/obs-log-dump
```

```
{  
  "log_dump": true  
}
```

## 响应示例

无

## 状态码

状态码	描述
204	Success。
400	BadRequest。

## 11.9.6 开启数据服务集群 LTS 日志转储

### 功能介绍

开启数据服务集群LTS日志转储。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/service/instances/{instance\_id}/lts-log-dump

表 11-344 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
instance_id	是	String	集群ID编号。



## 请求参数

表 11-345 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token，使用Token认证时必选。通过调用IAM服务的“获取用户Token”接口获取响应消息头中X-Subject-Token的值。
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
Dlm-Type	否	String	数据服务的版本类型，指定SHARED共享版或EXCLUSIVE专享版。
Content-Type	否	String	消息体的类型（格式），有Body体的情况下必选，没有Body体无需填写。如果请求消息体中含有中文字符，则需要通过charset=utf8指定中文字符集，例如取值为：application/json;charset=utf8。

表 11-346 请求 Body 参数

参数	是否必选	参数类型	描述
log_dump	否	Boolean	是否开启LTS日志转储功能，true表示开启，false表示关闭。
log_group_id	否	String	日志组ID。
log_group_name	否	String	日志组名称。
log_stream_id	否	String	日志流ID。
log_stream_name	否	String	日志流名称。

## 响应参数

状态码： 400

表 11-347 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

开启id为4f9366ce3e87860934c36d0ac027ad6a的集群LTS日志转储功能。

```
/v1/0833a5737480d53b2f250010d01a7b88/service/instances/4f9366ce3e87860934c36d0ac027ad6a/lts-log-dump  
{  
  "log_dump": true,  
  "log_group_id": "086d4d0b-89dc-468b-b0ff-37775d230eb6",  
  "log_group_name": "lts-group-mchuang",  
  "log_stream_id": "c4422c49-80d6-4825-bdf3-0bdf07ac564f",  
  "log_stream_name": "lts-topic-nc"  
}
```

## 响应示例

无

## 状态码

状态码	描述
204	Success。
400	BadRequest。

# 12 数据安全 API

## 12.1 权限管理接口

### 12.1.1 创建权限集

#### 功能介绍

创建权限集。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/{project\_id}/security/permission-sets

表 12-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

#### 请求参数

表 12-2 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

表 12-3 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	名称。
parent_id	否	String	父权限集id。
description	否	String	描述。
type	否	String	权限集类型, COMMON, MRS_MANAGED
managed_cluster_id	否	String	纳管角色所在集群id（仅纳管类权限集需要）。
managed_cluster_name	否	String	纳管角色所在集群名称（仅纳管类权限集需要）。
managed_role_name	否	String	纳管角色名称（仅纳管类权限集需要）。
manager_id	否	String	管理员id。
manager_name	否	String	管理员名称。
manager_type	否	String	管理员类型, 用户/用户组, USER, USER_GROUP

## 响应参数

状态码： 200

表 12-4 响应 Body 参数

参数	参数类型	描述
id	String	编号。
parent_id	String	父权限集id。
name	String	名称。
description	String	描述。

参数	参数类型	描述
type	String	权限集类型, COMMON, MRS_MANAGED
managed_cluster_id	String	纳管角色所在集群id (仅纳管类权限集需要)。
managed_cluster_name	String	纳管角色所在集群名称 (仅纳管类权限集需要)。
project_id	String	项目id。
domain_id	String	租户id。
instance_id	String	实例id。
manager_id	String	管理员id。
manager_name	String	管理员名称。
manager_type	String	管理员类型。
datasource_type	String	数据源类型。
sync_status	String	同步状态, UNKNOWN, NOT_SYNC, SYNCING, SYNC_SUCCESS, SYNC_FAIL
sync_msg	String	同步信息。
sync_time	Long	同步时间。
create_time	Long	创建时间。
create_user	String	创建者。
update_time	Long	更新时间。
update_user	String	更新者。

**状态码： 400**

**表 12-5 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets
```

```
{
  "name": "test_permission_set",
  "parent_id": "0",
  "description": "test permission set",
  "manager_id": "b43927a45a514e93ba70524b28923500",
  "manager_name": "common_user",
  "manager_type": "USER"
}
```

## 响应示例

状态码: 200

OK

```
{
  "create_time": 1698202688247,
  "create_user": "ei_dayu_test_01",
  "datasource_type": null,
  "description": "test permission set",
  "domain_id": "0833a5736980d53b0f22c0102ffcbfc0",
  "id": "fea96c90024711b8bf8d6886407b814b",
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "managed_cluster_id": null,
  "managed_cluster_name": null,
  "manager_id": "b43927a45a514e93ba70524b28923500",
  "manager_name": "common_user",
  "manager_type": "USER",
  "name": "test_permission_set",
  "parent_id": "0",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "sync_msg": null,
  "sync_status": "NOT_SYNC",
  "sync_time": null,
  "type": "COMMON",
  "update_time": 1698202688247,
  "update_user": "ei_dayu_test_01"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateSecurityPermissionSetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
    }
}
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();

CreateSecurityPermissionSetRequest request = new CreateSecurityPermissionSetRequest();
PermissionSetCreateDTO body = new PermissionSetCreateDTO();
body.withManagerType(PermissionSetCreateDTO.ManagerTypeEnum.fromValue("USER"));
body.withManagerName("common_user");
body.withManagerId("b43927a45a514e93ba70524b28923500");
body.withDescription("test permission set");
body.withParentId("0");
body.withName("test_permission_set");
request.withBody(body);
try {
    CreateSecurityPermissionSetResponse response = client.createSecurityPermissionSet(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecurityPermissionSetRequest()
        request.body = PermissionSetCreateDTO(
            manager_type="USER",
            manager_name="common_user",
```

```
        manager_id="b43927a45a514e93ba70524b28923500",
        description="test permission set",
        parent_id="0",
        name="test_permission_set"
    )
    response = client.create_security_permission_set(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityPermissionSetRequest{
        managerTypePermissionSetCreateDto:= model.GetPermissionSetCreateDtoManagerTypeEnum().USER
        managerNamePermissionSetCreateDto:= "common_user"
        managerIdPermissionSetCreateDto:= "b43927a45a514e93ba70524b28923500"
        descriptionPermissionSetCreateDto:= "test permission set"
        parentIdPermissionSetCreateDto:= "0"
        namePermissionSetCreateDto:= "test_permission_set"
        request.Body = &model.PermissionSetCreateDto{
            ManagerType: &managerTypePermissionSetCreateDto,
            ManagerName: &managerNamePermissionSetCreateDto,
            ManagerId: &managerIdPermissionSetCreateDto,
            Description: &descriptionPermissionSetCreateDto,
            ParentId: &parentIdPermissionSetCreateDto,
            Name: &namePermissionSetCreateDto,
        }
    }
    response, err := client.CreateSecurityPermissionSet(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```



## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.2 查询权限集列表

### 功能介绍

查询权限集列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/permission-sets

表 12-6 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 12-7 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页显示的条目数量。
offset	否	Integer	偏移量，表示从此偏移量开始查询，该值大于等于0。
name	否	String	名称。
parent_id	否	String	父权限集id。

参数	是否必选	参数类型	描述
type_filter	否	String	权限集类型过滤, TOP_PERMISSION_SET, SUB_PERMISSION_SET, ALL_PERMISSION_SET。
manager_id	否	String	管理员id。
manager_name	否	String	管理员名称。
manager_type	否	String	管理员类型, USER, USER_GROUP。
datasource_type	否	String	数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul>
sync_status	否	String	同步状态, UNKNOWN, NOT_SYNC, SYNCING, SYNC_SUCCESS, SYNC_FAIL
order_by	否	String	排序参数, NAME, CREATE_TIME, UPDATE_TIME。
order_by_asc	否	Boolean	是否升序（仅指定排序参数时有效）。

## 请求参数

表 12-8 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token, 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

## 响应参数

状态码： 200

表 12-9 响应 Body 参数

参数	参数类型	描述
total	Integer	总条数。
permission_sets	Array of <a href="#">PermissionSet</a> objects	权限集列表。

表 12-10 PermissionSet

参数	参数类型	描述
id	String	编号。
parent_id	String	父权限集id。
name	String	名称。
description	String	描述。
type	String	权限集类型, COMMON, MRS_MANAGED
managed_cluster_id	String	纳管角色所在集群id（仅纳管类权限集需要）。
managed_cluster_name	String	纳管角色所在集群名称（仅纳管类权限集需要）。
project_id	String	项目id。
domain_id	String	租户id。
instance_id	String	实例id。
manager_id	String	管理员id。
manager_name	String	管理员名称。
manager_type	String	管理员类型。
datasource_type	String	数据源类型。
sync_status	String	同步状态, UNKNOWN, NOT_SYNC, SYNCING, SYNC_SUCCESS, SYNC_FAIL
sync_msg	String	同步信息。
sync_time	Long	同步时间。
create_time	Long	创建时间。
create_user	String	创建者。

参数	参数类型	描述
update_time	Long	更新时间。
update_user	String	更新者。

状态码： 400

表 12-11 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets?
offset=0&limit=10&type_filter=TOP_PERMISSION_SET
```

## 响应示例

状态码： 200

OK

```
{
  "permission_sets": [ {
    "create_time": 1698202688000,
    "create_user": "ei_dayu_test_01",
    "datasource_type": null,
    "description": "test permission set",
    "domain_id": "0833a5736980d53b0f22c0102ffcbfc0",
    "id": "fea96c90024711b8bf8d6886407b814b",
    "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
    "managed_cluster_id": null,
    "managed_cluster_name": null,
    "manager_id": "b43927a45a514e93ba70524b28923500",
    "manager_name": "common_user",
    "manager_type": "USER",
    "name": "test_permission_set",
    "parent_id": "0",
    "project_id": "0833a5737480d53b2f25c010dc1a7b88",
    "sync_msg": null,
    "sync_status": "NOT_SYNC",
    "sync_time": null,
    "type": "COMMON",
    "update_time": 1698202688000,
    "update_user": "ei_dayu_test_01"
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityPermissionSetsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityPermissionSetsRequest request = new ListSecurityPermissionSetsRequest();
        try {
            ListSecurityPermissionSetsResponse response = client.listSecurityPermissionSets(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSecurityPermissionSetsRequest()
    response = client.list_security_permission_sets(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityPermissionSetsRequest{}
    response, err := client.ListSecurityPermissionSets(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.3 查询权限集

### 功能介绍

查询权限集。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/permission-sets/{permission\_set\_id}

表 12-12 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
permission_set_id	是	String	权限集id。

### 请求参数

表 12-13 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

## 响应参数

状态码： 200

表 12-14 响应 Body 参数

参数	参数类型	描述
id	String	编号。
parent_id	String	父权限集id。
name	String	名称。
description	String	描述。
type	String	权限集类型, COMMON, MRS_MANAGED
managed_cluster_id	String	纳管角色所在集群id（仅纳管类权限集需要）。
managed_cluster_name	String	纳管角色所在集群名称（仅纳管类权限集需要）。
project_id	String	项目id。
domain_id	String	租户id。
instance_id	String	实例id。
manager_id	String	管理员id。
manager_name	String	管理员名称。
manager_type	String	管理员类型。
datasource_type	String	数据源类型。
sync_status	String	同步状态, UNKNOWN, NOT_SYNC, SYNCING, SYNC_SUCCESS, SYNC_FAIL
sync_msg	String	同步信息。
sync_time	Long	同步时间。
create_time	Long	创建时间。
create_user	String	创建者。
update_time	Long	更新时间。
update_user	String	更新者。

状态码： 400



表 12-15 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/fea96c90024711b8bf8d6886407b814b
```

## 响应示例

状态码： 200

OK

```
{
  "create_time": 1698202688247,
  "create_user": "ei_dayu_test_01",
  "datasource_type": null,
  "description": "test permission set",
  "domain_id": "0833a5736980d53b0f22c0102ffc0",
  "id": "fea96c90024711b8bf8d6886407b814b",
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "managed_cluster_id": null,
  "managed_cluster_name": null,
  "manager_id": "b43927a45a514e93ba70524b28923500",
  "manager_name": "common_user",
  "manager_type": "USER",
  "name": "test_permission_set",
  "parent_id": "0",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "sync_msg": null,
  "sync_status": "NOT_SYNC",
  "sync_time": null,
  "type": "COMMON",
  "update_time": 1698202688247,
  "update_user": "ei_dayu_test_01"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowSecurityPermissionSetSolution {

    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowSecurityPermissionSetRequest request = new ShowSecurityPermissionSetRequest();
request.withPermissionSetId("{permission_set_id}");
try {
    ShowSecurityPermissionSetResponse response = client.showSecurityPermissionSet(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecurityPermissionSetRequest()
        request.permission_set_id = "{permission_set_id}"
        response = client.show_security_permission_set(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSecurityPermissionSetRequest{}
    request.PermissionSetId = "{permission_set_id}"
    response, err := client.ShowSecurityPermissionSet(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.4 删除权限集

### 功能介绍

删除权限集。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/{project\_id}/security/permission-sets/{permission\_set\_id}

表 12-16 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
permission_set_id	是	String	权限集id。

### 请求参数

表 12-17 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

### 响应参数

状态码： 400

表 12-18 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/fea96c90024711b8bf8d6886407b814b
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteSecurityPermissionSetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteSecurityPermissionSetRequest request = new DeleteSecurityPermissionSetRequest();
        request.withPermissionSetId("{permission_set_id}");
        try {
            DeleteSecurityPermissionSetResponse response = client.deleteSecurityPermissionSet(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecurityPermissionSetRequest()
        request.permission_set_id = "{permission_set_id}"
        response = client.delete_security_permission_set(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```

        WithCredential(auth).
        Build())

    request := &model.DeleteSecurityPermissionSetRequest{}
    request.PermissionSetId = "{permission_set_id}"
    response, err := client.DeleteSecurityPermissionSet(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	No Content
400	Bad Request

## 12.1.5 更新权限集

### 功能介绍

更新权限集。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/security/permission-sets/{permission\_set\_id}

表 12-19 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
permission_set_id	是	String	权限集id。

## 请求参数

表 12-20 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

表 12-21 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	名称。
parent_id	否	String	父权限集id。
description	否	String	描述。
type	否	String	权限集类型, COMMON, MRS_MANAGED
managed_cluster_id	否	String	纳管角色所在集群id（仅纳管类权限集需要）。
managed_cluster_name	否	String	纳管角色所在集群名称（仅纳管类权限集需要）。
managed_role_name	否	String	纳管角色名称（仅纳管类权限集需要）。
manager_id	否	String	管理员id。
manager_name	否	String	管理员名称。
manager_type	否	String	管理员类型, 用户/用户组, USER, USER_GROUP

## 响应参数

状态码： 200



表 12-22 响应 Body 参数

参数	参数类型	描述
id	String	编号。
parent_id	String	父权限集id。
name	String	名称。
description	String	描述。
type	String	权限集类型, COMMON, MRS_MANAGED
managed_cluster_id	String	纳管角色所在集群id（仅纳管类权限集需要）。
managed_cluster_name	String	纳管角色所在集群名称（仅纳管类权限集需要）。
project_id	String	项目id。
domain_id	String	租户id。
instance_id	String	实例id。
manager_id	String	管理员id。
manager_name	String	管理员名称。
manager_type	String	管理员类型。
datasource_type	String	数据源类型。
sync_status	String	同步状态,UNKNOWN,NOT_SYNC,SYNCING,SYNC_SUCCESS,SYNC_FAIL
sync_msg	String	同步信息。
sync_time	Long	同步时间。
create_time	Long	创建时间。
create_user	String	创建者。
update_time	Long	更新时间。
update_user	String	更新者。

状态码： 400

表 12-23 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/fea96c90024711b8bf8d6886407b814b
{
  "name": "test_permission_set",
  "parent_id": "0",
  "description": "test permission set",
  "manager_id": "b43927a45a514e93ba70524b28923500",
  "manager_name": "common_user",
  "manager_type": "USER"
}
```

## 响应示例

状态码： 200

OK

```
{
  "create_time": 1698202688247,
  "create_user": "ei_dayu_test_01",
  "datasource_type": null,
  "description": "test permission set",
  "domain_id": "0833a5736980d53b0f22c0102ffc0",
  "id": "fea96c90024711b8bf8d6886407b814b",
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "managed_cluster_id": null,
  "managed_cluster_name": null,
  "manager_id": "b43927a45a514e93ba70524b28923500",
  "manager_name": "common_user",
  "manager_type": "USER",
  "name": "test_permission_set",
  "parent_id": "0",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "sync_msg": null,
  "sync_status": "NOT_SYNC",
  "sync_time": null,
  "type": "COMMON",
  "update_time": 1698202688247,
  "update_user": "ei_dayu_test_01"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateSecurityPermissionSetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSecurityPermissionSetRequest request = new UpdateSecurityPermissionSetRequest();
        request.withPermissionSetId("{permission_set_id}");
        PermissionSetCreateDTO body = new PermissionSetCreateDTO();
        body.withManagerType(PermissionSetCreateDTO.ManagerTypeEnum.fromValue("USER"));
        body.withManagerName("common_user");
        body.withManagerId("b43927a45a514e93ba70524b28923500");
        body.withDescription("test permission set");
        body.withParentId("0");
        body.withName("test_permission_set");
        request.withBody(body);
        try {
            UpdateSecurityPermissionSetResponse response = client.updateSecurityPermissionSet(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateSecurityPermissionSetRequest()
    request.permission_set_id = "{permission_set_id}"
    request.body = PermissionSetCreateDTO(
        manager_type="USER",
        manager_name="common_user",
        manager_id="b43927a45a514e93ba70524b28923500",
        description="test permission set",
        parent_id="0",
        name="test_permission_set"
    )
    response = client.update_security_permission_set(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecurityPermissionSetRequest{}
    request.PermissionSetId = "{permission_set_id}"
    managerTypePermissionSetCreateDto := model.GetPermissionSetCreateDtoManagerTypeEnum().USER
    managerNamePermissionSetCreateDto := "common_user"
```

```
managerIdPermissionSetCreateDto:= "b43927a45a514e93ba70524b28923500"
descriptionPermissionSetCreateDto:= "test permission set"
parentIdPermissionSetCreateDto:= "0"
namePermissionSetCreateDto:= "test_permission_set"
request.Body = &model.PermissionSetCreateDto{
    ManagerType: &managerTypePermissionSetCreateDto,
    ManagerName: &managerNamePermissionSetCreateDto,
    ManagerId: &managerIdPermissionSetCreateDto,
    Description: &descriptionPermissionSetCreateDto,
    ParentId: &parentIdPermissionSetCreateDto,
    Name: &namePermissionSetCreateDto,
}
response, err := client.UpdateSecurityPermissionSet(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.6 添加权限集成员

### 功能介绍

添加权限集成员。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-sets/{permission\_set\_id}/members

表 12-24 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

参数	是否必选	参数类型	描述
permission_set_id	是	String	权限集id。

## 请求参数

表 12-25 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

表 12-26 请求 Body 参数

参数	是否必选	参数类型	描述
member_type	否	String	成员类型，用户/用户组/工作空间角色，USER, USER_GROUP, WORKSPACE_ROLE
member_id	否	String	成员id。
member_name	否	String	成员名称。
workspace	否	String	工作空间。

## 响应参数

状态码： 200

表 12-27 响应 Body 参数

参数	参数类型	描述
id	String	id。
permission_set_id	String	权限集id。
project_id	String	项目id。

参数	参数类型	描述
instance_id	String	实例id。
member_type	String	成员类型, 用户/用户组/工作空间角色(废弃)/集群角色, USER, USER_GROUP, WORKSPACE_ROLE, CLUSTER_ROLE
member_id	String	成员id。
member_name	String	成员名称。
member_status	String	成员状态, NORMAL, UNFINISHED
workspace	String	工作空间(仅工作空间角色需要)。
cluster_type	String	集群类型(仅集群角色需要), MRS, DWS, DLI。
cluster_id	String	集群id(仅集群角色需要)。
cluster_name	String	集群名称(仅集群角色需要)。
create_time	Long	创建时间。
create_user	String	创建者。
deadline	Long	到期时间。

状态码: 400

表 12-28 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/members
{
  "member_id": "b43927a45a514e93ba70524b28923500",
  "member_name": "common_user",
  "member_type": "USER",
  "workspace": null
}
```

## 响应示例

状态码: 200

OK

```
{
  "cluster_id" : null,
  "cluster_name" : null,
  "cluster_type" : null,
  "create_time" : 1698204793000,
  "create_user" : "ei_dayu_test_01",
  "deadline" : null,
  "id" : "d9ff4b06db43e3d4f81de1e60077a480",
  "instance_id" : "dd97167b873d4a79b2aad54d4370a3bc",
  "member_id" : "b43927a45a514e93ba70524b28923500",
  "member_name" : "common_user",
  "member_status" : "NORMAL",
  "member_type" : "USER",
  "permission_set_id" : "fea96c90024711b8bf8d6886407b814b",
  "project_id" : "0833a5737480d53b2f25c010dc1a7b88",
  "workspace" : null
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class CreateSecurityPermissionSetMemberSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateSecurityPermissionSetMemberRequest request = new
        CreateSecurityPermissionSetMemberRequest();
        request.withPermissionSetId("{permission_set_id}");
        PermissionSetMemberCreateDTO body = new PermissionSetMemberCreateDTO();
        body.withMemberName("common_user");
        body.withMemberId("b43927a45a514e93ba70524b28923500");
        body.withMemberType(PermissionSetMemberCreateDTO.MemberTypeEnum.fromValue("USER"));
        request.withBody(body);
        try {
            CreateSecurityPermissionSetMemberResponse response =
            client.createSecurityPermissionSetMember(request);
        }
    }
}
```



```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecurityPermissionSetMemberRequest()
        request.permission_set_id = "{permission_set_id}"
        request.body = PermissionSetMemberCreateDTO(
            member_name="common_user",
            member_id="b43927a45a514e93ba70524b28923500",
            member_type="USER"
        )
        response = client.create_security_permission_set_member(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityPermissionSetMemberRequest{}
    request.PermissionSetId = "{permission_set_id}"
    memberNamePermissionSetMemberCreateDto := "common_user"
    memberIdPermissionSetMemberCreateDto := "b43927a45a514e93ba70524b28923500"
    memberTypePermissionSetMemberCreateDto :=
model.GetPermissionSetMemberCreateDtoMemberTypeEnum().USER
    request.Body = &model.PermissionSetMemberCreateDto{
        MemberName: &memberNamePermissionSetMemberCreateDto,
        MemberId: &memberIdPermissionSetMemberCreateDto,
        MemberType: &memberTypePermissionSetMemberCreateDto,
    }
    response, err := client.CreateSecurityPermissionSetMember(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.7 查询权限集成员列表

### 功能介绍

查询权限集成员列表。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/security/permission-sets/{permission\_set\_id}/members

表 12-29 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
permission_set_id	是	String	权限集id。

表 12-30 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页显示的条目数量。
offset	否	Integer	偏移量，表示从此偏移量开始查询，该值大于等于0。
member_name	否	String	成员名称。
member_type	否	String	成员类型, USER, USER_GROUP, WORKSPACE_ROLE。
order_by	否	String	排序参数, CREATE_TIME, MEMBER_NAME。
order_by_asc	否	Boolean	是否升序（仅指定排序参数时有效）。

## 请求参数

表 12-31 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

## 响应参数

状态码： 200

表 12-32 响应 Body 参数

参数	参数类型	描述
total	Integer	总条数。
permission_set_members	Array of <a href="#">PermissionSetMember</a> objects	成员列表。

表 12-33 PermissionSetMember

参数	参数类型	描述
id	String	id。
permission_set_id	String	权限集id。
project_id	String	项目id。
instance_id	String	实例id。
member_type	String	成员类型, 用户/用户组/工作空间角色(废弃)/集群角色, USER, USER_GROUP, WORKSPACE_ROLE, CLUSTER_ROLE
member_id	String	成员id。
member_name	String	成员名称。
member_status	String	成员状态, NORMAL, UNFINISHED
workspace	String	工作空间(仅工作空间角色需要)。
cluster_type	String	集群类型(仅集群角色需要), MRS, DWS, DLI。
cluster_id	String	集群id(仅集群角色需要)。

参数	参数类型	描述
cluster_name	String	集群名称(仅集群角色需要)。
create_time	Long	创建时间。
create_user	String	创建者。
deadline	Long	到期时间。

状态码： 400

表 12-34 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/members
```

## 响应示例

状态码： 200

OK

```
{
  "permission_set_members": [ {
    "cluster_id": null,
    "cluster_name": null,
    "cluster_type": null,
    "create_time": 1698204793000,
    "create_user": "ei_dayu_test_01",
    "deadline": null,
    "id": "d9ff4b06db43e3d4f81de1e60077a480",
    "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
    "member_id": "b43927a45a514e93ba70524b28923500",
    "member_name": "common_user",
    "member_status": "NORMAL",
    "member_type": "USER",
    "permission_set_id": "fea96c90024711b8bf8d6886407b814b",
    "project_id": "0833a5737480d53b2f25c010dc1a7b88",
    "workspace": null
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityPermissionSetMembersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityPermissionSetMembersRequest request = new ListSecurityPermissionSetMembersRequest();
        request.withPermissionSetId("{permission_set_id}");
        try {
            ListSecurityPermissionSetMembersResponse response =
                client.listSecurityPermissionSetMembers(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSecurityPermissionSetMembersRequest()
    request.permission_set_id = "{permission_set_id}"
    response = client.list_security_permission_set_members(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityPermissionSetMembersRequest{}
    request.PermissionSetId = "{permission_set_id}"
    response, err := client.ListSecurityPermissionSetMembers(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.8 批量删除权限集成员

### 功能介绍

批量删除权限集成员。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-sets/{permission\_set\_id}/members/  
batch-delete

表 12-35 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
permission_set_id	是	String	权限集id。

### 请求参数

表 12-36 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。



参数	是否必选	参数类型	描述
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

表 12-37 请求 Body 参数

参数	是否必选	参数类型	描述
dw_id	否	String	数据连接id。
ids	否	Array of strings	id列表。

## 响应参数

状态码： 400

表 12-38 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/members/batch-delete
{
  "dw_id": null,
  "ids": [ "d9ff4b06db43e3d4f81de1e60077a480" ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteSecurityPermissionSetMembersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchDeleteSecurityPermissionSetMembersRequest request = new
        BatchDeleteSecurityPermissionSetMembersRequest();
        request.withPermissionSetId("{permission_set_id}");
        BatchDeleteBaseDTO body = new BatchDeleteBaseDTO();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("d9ff4b06db43e3d4f81de1e60077a480");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            BatchDeleteSecurityPermissionSetMembersResponse response =
            client.batchDeleteSecurityPermissionSetMembers(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = BatchDeleteSecurityPermissionSetMembersRequest()
    request.permission_set_id = "{permission_set_id}"
    listIdsbody = [
        "d9ff4b06db43e3d4f81de1e60077a480"
    ]
    request.body = BatchDeleteBaseDTO(
        ids=listIdsbody
    )
    response = client.batch_delete_security_permission_set_members(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteSecurityPermissionSetMembersRequest{}
    request.PermissionSetId = "{permission_set_id}"
    var listIdsbody = []string{
        "d9ff4b06db43e3d4f81de1e60077a480",
    }
```

```

    }
    request.Body = &model.BatchDeleteBaseDto{
        Ids: &listIdsbody,
    }
    response, err := client.BatchDeleteSecurityPermissionSetMembers(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	No Content
400	Bad Request

## 12.1.9 添加权限集的权限

### 功能介绍

添加权限集的权限。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-sets/{permission\_set\_id}/permissions

表 12-39 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
permission_set_id	是	String	权限集id。

## 请求参数

表 12-40 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

表 12-41 请求 Body 参数

参数	是否必选	参数类型	描述
dw_id	否	String	数据连接id。
permission_type	否	String	权限类型, DENY, ALLOW
permission_actions	否	Array of strings	权限操作列表。
cluster_id	否	String	集群id。
cluster_name	否	String	集群名称。
datasource_type	否	String	数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul>
database_name	否	String	数据库名称。
schema_name	否	String	模式名称。
namespace	否	String	命名空间。无效参数，待下线。
table_name	否	String	表名称。
column_name	否	String	列名称。
row_level_security	否	String	行级策略。无效参数，待下线。
url	否	String	url路径名称, MRS存算分离或者HIVE指定location场景下使用。

## 响应参数

状态码： 200

表 12-42 响应 Body 参数

参数	参数类型	描述
id	String	id。
permission_set_id	String	权限集id。
project_id	String	项目id。
instance_id	String	实例id。
permission_type	String	权限类型, DENY, ALLOW
permission_action	String	权限操作列表。
permission_actions	Array of strings	权限操作列表。
permission_action_code	Integer	权限操作编码, 位图。
cluster_id	String	集群id。
cluster_name	String	集群名称。
datasource_type	String	数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul>
database_name	String	数据库名称。
schema_name	String	模式名称。
namespace	String	命名空间。
table_name	String	表名称。
column_name	String	列名称。
row_level_security	String	行级策略。
sync_status	String	同步状态, UNKNOWN, NOT_SYNC, SYNCING, SYNC_SUCCESS, SYNC_FAIL
sync_msg	String	同步信息。

参数	参数类型	描述
url	String	url路径名称。

状态码： 400

表 12-43 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/fea96c90024711b8bf8d6886407b814b/permissions
{
  "permission_type": "ALLOW",
  "cluster_id": "4367f185-79d4-41ae-afd5-54d55176aa68",
  "cluster_name": "mrs_3x_autotest_do_not_del",
  "datasource_type": "HIVE",
  "database_name": "default",
  "dw_id": "88055a9005194bdfa4377714ca166369",
  "permission_actions": [ "SELECT" ],
  "table_name": "*",
  "column_name": "*"
}
```

## 响应示例

状态码： 200

OK

```
{
  "cluster_id": "4367f185-79d4-41ae-afd5-54d55176aa68",
  "cluster_name": "mrs_3x_autotest_do_not_del",
  "column_name": "*",
  "database_name": "default",
  "datasource_type": "HIVE",
  "id": "511fe36e2cc0e33a094a8bdaa4b73e55",
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "namespace": null,
  "permission_action": "SELECT",
  "permission_action_code": 2,
  "permission_actions": [ "SELECT" ],
  "permission_set_id": "fea96c90024711b8bf8d6886407b814b",
  "permission_type": "ALLOW",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "row_level_security": null,
  "schema_name": null,
  "sync_msg": null,
  "sync_status": "NOT_SYNC",
  "table_name": "*",
  "url": null
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateSecurityPermissionSetPermissionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateSecurityPermissionSetPermissionRequest request = new
        CreateSecurityPermissionSetPermissionRequest();
        request.withPermissionSetId("{permission_set_id}");
        PermissionSetPermissionCreateDTO body = new PermissionSetPermissionCreateDTO();
        List<PermissionSetPermissionCreateDTO.PermissionActionsEnum> listbodyPermissionActions = new
        ArrayList<>();

        listbodyPermissionActions.add(PermissionSetPermissionCreateDTO.PermissionActionsEnum.fromValue("SELE
        CT"));
        body.withColumnName("**");
        body.withTableName("**");
        body.withDatabaseName("default");

        body.withDatasourceType(PermissionSetPermissionCreateDTO.DatasourceTypeEnum.fromValue("HIVE"));
        body.withClusterName("mrs_3x_autotest_do_not_del");
        body.withClusterId("4367f185-79d4-41ae-afd5-54d55176aa68");
        body.withPermissionActions(listbodyPermissionActions);

        body.withPermissionType(PermissionSetPermissionCreateDTO.PermissionTypeEnum.fromValue("ALLOW"));
        body.withDwId("88055a9005194bdfa4377714ca166369");
        request.withBody(body);
        try {
            CreateSecurityPermissionSetPermissionResponse response =
            client.createSecurityPermissionSetPermission(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```



```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecurityPermissionSetPermissionRequest()
        request.permission_set_id = "{permission_set_id}"
        listPermissionActionsbody = [
            "SELECT"
        ]
        request.body = PermissionSetPermissionCreateDTO(
            column_name="*",
            table_name="*",
            database_name="default",
            datasource_type="HIVE",
            cluster_name="mrs_3x_autotest_do_not_del",
            cluster_id="4367f185-79d4-41ae-afd5-54d55176aa68",
            permission_actions=listPermissionActionsbody,
            permission_type="ALLOW",
            dw_id="88055a9005194bdfa4377714ca166369"
        )
        response = client.create_security_permission_set_permission(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityPermissionSetPermissionRequest{}
    request.PermissionSetId = "{permission_set_id}"
    var listPermissionActionsbody = []model.PermissionSetPermissionCreateDtoPermissionActions{
        model.GetPermissionSetPermissionCreateDtoPermissionActionsEnum().SELECT,
    }
    columnNamePermissionSetPermissionCreateDto:= ""
    tableNamePermissionSetPermissionCreateDto:= ""
    databaseNamePermissionSetPermissionCreateDto:= "default"
    datasourceTypePermissionSetPermissionCreateDto:=
model.GetPermissionSetPermissionCreateDtoDatasourceTypeEnum().HIVE
    clusterNamePermissionSetPermissionCreateDto:= "mrs_3x_autotest_do_not_del"
    clusterIdPermissionSetPermissionCreateDto:= "4367f185-79d4-41ae-afd5-54d55176aa68"
    permissionTypePermissionSetPermissionCreateDto:=
model.GetPermissionSetPermissionCreateDtoPermissionTypeEnum().ALLOW
    dwldPermissionSetPermissionCreateDto:= "88055a9005194bdfa4377714ca166369"
    request.Body = &model.PermissionSetPermissionCreateDto{
        ColumnName: &columnNamePermissionSetPermissionCreateDto,
        TableName: &tableNamePermissionSetPermissionCreateDto,
        DatabaseName: &databaseNamePermissionSetPermissionCreateDto,
        DatasourceType: &datasourceTypePermissionSetPermissionCreateDto,
        ClusterName: &clusterNamePermissionSetPermissionCreateDto,
        ClusterId: &clusterIdPermissionSetPermissionCreateDto,
        PermissionActions: &listPermissionActionsbody,
        PermissionType: &permissionTypePermissionSetPermissionCreateDto,
        Dwld: &dwldPermissionSetPermissionCreateDto,
    }
    response, err := client.CreateSecurityPermissionSetPermission(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.10 查询权限集的权限列表

### 功能介绍

查询权限集的权限列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/permission-sets/{permission\_set\_id}/permissions

表 12-44 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
permission_set_id	是	String	权限集id。

表 12-45 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页显示的条目数量。
offset	否	Integer	偏移量，表示从此偏移量开始查询，该值大于等于0。
permission_type	否	String	权限类型,DENY,ALLOW
permission_action	否	String	权限操作,ALL,SELECT,UPDATE,CREATE,DROP,ALTER,INDEX,LOCK,READ,WRITE
cluster_id	否	String	集群ID，获取方法请参见 <a href="#">查询单个数据连接信息</a> 。

参数	是否必选	参数类型	描述
cluster_name	否	String	集群名称。
datasource_type	否	String	数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul>
database_name	否	String	数据库名称。
table_name	否	String	表名称。
column_name	否	String	列名称。
sync_status	否	String	同步状态,UNKNOWN,NOT_SYNC,SYNCING,SYNC_SUCCESS,SYNC_FAIL
order_by	否	String	排序参数, CLUSTER_NAME, DATABASE_NAME。
order_by_asc	否	Boolean	是否升序（仅指定排序参数时有效）。

## 请求参数

表 12-46 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

## 响应参数

状态码： 200

表 12-47 响应 Body 参数

参数	参数类型	描述
total	Integer	总条数。

参数	参数类型	描述
permissions	Array of <a href="#">PermissionSetPermission</a> objects	权限列表。

表 12-48 PermissionSetPermission

参数	参数类型	描述
id	String	id。
permission_set_id	String	权限集id。
project_id	String	项目id。
instance_id	String	实例id。
permission_type	String	权限类型, DENY, ALLOW
permission_action	String	权限操作列表。
permission_actions	Array of strings	权限操作列表。
permission_action_code	Integer	权限操作编码, 位图。
cluster_id	String	集群id。
cluster_name	String	集群名称。
datasource_type	String	数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul>
database_name	String	数据库名称。
schema_name	String	模式名称。
namespace	String	命名空间。
table_name	String	表名称。
column_name	String	列名称。
row_level_security	String	行级策略。

参数	参数类型	描述
sync_status	String	同步状态,UNKNOWN,NOT_SYNC,SYNCING,SYNC_SUCCESS,SYNC_FAIL
sync_msg	String	同步信息。
url	String	url路径名称。

状态码： 400

表 12-49 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/fea96c90024711b8bf8d6886407b814b/permissions?offset=0&limit=10
```

## 响应示例

状态码： 200

OK

```
{
  "permissions": [ {
    "cluster_id": "4367f185-79d4-41ae-afd5-54d55176aa68",
    "cluster_name": "mrs_3x_autotest_do_not_del",
    "column_name": "*",
    "database_name": "default",
    "datasource_type": "HIVE",
    "id": "511fe36e2cc0e33a094a8bdaa4b73e55",
    "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
    "namespace": null,
    "permission_action": "SELECT",
    "permission_action_code": 2,
    "permission_actions": [ "SELECT" ],
    "permission_set_id": "fea96c90024711b8bf8d6886407b814b",
    "permission_type": "ALLOW",
    "project_id": "0833a5737480d53b2f25c010dc1a7b88",
    "row_level_security": null,
    "schema_name": null,
    "sync_msg": null,
    "sync_status": "NOT_SYNC",
    "table_name": "*"
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityPermissionSetPermissionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityPermissionSetPermissionsRequest request = new
        ListSecurityPermissionSetPermissionsRequest();
        request.withPermissionSetId("{permission_set_id}");
        try {
            ListSecurityPermissionSetPermissionsResponse response =
            client.listSecurityPermissionSetPermissions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecurityPermissionSetPermissionsRequest()
        request.permission_set_id = "{permission_set_id}"
        response = client.list_security_permission_set_permissions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityPermissionSetPermissionsRequest{}
    request.PermissionSetId = "{permission_set_id}"
    response, err := client.ListSecurityPermissionSetPermissions(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```



```
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.11 删除权限集的权限

### 功能介绍

删除权限集的权限。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-sets/{permission\_set\_id}/permissions/  
batch-delete

表 12-50 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。
permission_set_id	是	String	权限集id。

## 请求参数

表 12-51 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

表 12-52 请求 Body 参数

参数	是否必选	参数类型	描述
dw_id	否	String	数据连接id。
ids	否	Array of strings	id列表。

## 响应参数

状态码： 400

表 12-53 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/fea96c90024711b8bf8d6886407b814b/permissions/batch-delete
{
  "ids" : [ "511fe36e2cc0e33a094a8bdaa4b73e55" ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteSecurityPermissionSetPermissionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchDeleteSecurityPermissionSetPermissionsRequest request = new
        BatchDeleteSecurityPermissionSetPermissionsRequest();
        request.withPermissionSetId("{permission_set_id}");
        BatchDeleteBaseDTO body = new BatchDeleteBaseDTO();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("511fe36e2cc0e33a094a8bdaa4b73e55");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            BatchDeleteSecurityPermissionSetPermissionsResponse response =
            client.batchDeleteSecurityPermissionSetPermissions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteSecurityPermissionSetPermissionsRequest()
        request.permission_set_id = "{permission_set_id}"
        listIdsbody = [
            "511fe36e2cc0e33a094a8bdaa4b73e55"
        ]
        request.body = BatchDeleteBaseDTO(
            ids=listIdsbody
        )
        response = client.batch_delete_security_permission_set_permissions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```

        WithCredential(auth).
        Build()

        request := &model.BatchDeleteSecurityPermissionSetPermissionsRequest{}
        request.PermissionSetId = "{permission_set_id}"
        var listIdsbody = []string{
            "511fe36e2cc0e33a094a8bdaa4b73e55",
        }
        request.Body = &model.BatchDeleteBaseDto{
            Ids: &listIdsbody,
        }
        response, err := client.BatchDeleteSecurityPermissionSetPermissions(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
204	No Content
400	Bad Request

## 12.1.12 更新权限集的权限

### 功能介绍

更新权限集的权限。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/security/permission-sets/{permission\_set\_id}/permissions/{permission\_id}

表 12-54 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

参数	是否必选	参数类型	描述
permission_set_id	是	String	权限集id。
permission_id	是	String	权限id。

## 请求参数

表 12-55 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

表 12-56 请求 Body 参数

参数	是否必选	参数类型	描述
dw_id	否	String	数据连接id。
permission_actions	否	Array of strings	权限操作列表。

## 响应参数

状态码： 200

表 12-57 响应 Body 参数

参数	参数类型	描述
id	String	id。
permission_set_id	String	权限集id。
project_id	String	项目id。
instance_id	String	实例id。
permission_type	String	权限类型, DENY, ALLOW

参数	参数类型	描述
permission_action	String	权限操作列表。
permission_actions	Array of strings	权限操作列表。
permission_action_code	Integer	权限操作编码, 位图。
cluster_id	String	集群id。
cluster_name	String	集群名称。
datasource_type	String	数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul>
database_name	String	数据库名称。
schema_name	String	模式名称。
namespace	String	命名空间。
table_name	String	表名称。
column_name	String	列名称。
row_level_security	String	行级策略。
sync_status	String	同步状态, UNKNOWN, NOT_SYNC, SYNCING, SYNC_SUCCESS, SYNC_FAIL
sync_msg	String	同步信息。
url	String	url路径名称。

状态码: 400

表 12-58 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/fea96c90024711b8bf8d6886407b814b/permissions/511fe36e2cc0e33a094a8bdaa4b73e55  
  
{  
  "permission_actions": [ "SELECT" ]  
}
```

## 响应示例

**状态码: 200**

OK

```
{  
  "cluster_id": "4367f185-79d4-41ae-afd5-54d55176aa68",  
  "cluster_name": "mrs_3x_autotest_do_not_del",  
  "column_name": "*",  
  "database_name": "default",  
  "datasource_type": "HIVE",  
  "id": "511fe36e2cc0e33a094a8bdaa4b73e55",  
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",  
  "namespace": null,  
  "permission_action": "SELECT",  
  "permission_action_code": 2,  
  "permission_actions": [ "SELECT" ],  
  "permission_set_id": "fea96c90024711b8bf8d6886407b814b",  
  "permission_type": "ALLOW",  
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",  
  "row_level_security": null,  
  "schema_name": null,  
  "sync_msg": null,  
  "sync_status": "NOT_SYNC",  
  "table_name": "*"   
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class UpdateSecurityPermissionSetPermissionSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
    }  
}
```



```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();

UpdateSecurityPermissionSetPermissionRequest request = new
UpdateSecurityPermissionSetPermissionRequest();
request.withPermissionSetId("{permission_set_id}");
request.withPermissionId("{permission_id}");
PermissionSetPermissionUpdateDTO body = new PermissionSetPermissionUpdateDTO();
List<PermissionSetPermissionUpdateDTO.PermissionActionsEnum> listbodyPermissionActions = new
ArrayList<>();

listbodyPermissionActions.add(PermissionSetPermissionUpdateDTO.PermissionActionsEnum.fromValue("SELECT"));
body.withPermissionActions(listbodyPermissionActions);
request.withBody(body);
try {
    UpdateSecurityPermissionSetPermissionResponse response =
client.updateSecurityPermissionSetPermission(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = UpdateSecurityPermissionSetPermissionRequest()
request.permission_set_id = "{permission_set_id}"
request.permission_id = "{permission_id}"
listPermissionActionsbody = [
    "SELECT"
]
request.body = PermissionSetPermissionUpdateDTO(
    permission_actions=listPermissionActionsbody
)
response = client.update_security_permission_set_permission(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecurityPermissionSetPermissionRequest{
        request.PermissionSetId = "{permission_set_id}"
        request.PermissionId = "{permission_id}"
        var listPermissionActionsbody = []model.PermissionSetPermissionUpdateDtoPermissionActions{
            model.GetPermissionSetPermissionUpdateDtoPermissionActionsEnum().SELECT,
        }
        request.Body = &model.PermissionSetPermissionUpdateDto{
            PermissionActions: &listPermissionActionsbody,
        }
        response, err := client.UpdateSecurityPermissionSetPermission(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.13 查询数据源可配置权限

### 功能介绍

查询数据源可配置权限。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/permission-sets/datasource/configurations

表 12-59 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 12-60 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

## 响应参数

状态码： 200

表 12-61 响应 Body 参数

参数	参数类型	描述
configurations	Array of <b>PermissionConfiguration</b> objects	数据源操作权限列表。

表 12-62 PermissionConfiguration

参数	参数类型	描述
datasource_type	String	数据源类型。
permission_types	Array of strings	数据源操作权限类型。
permission_actions	Array of strings	数据源支持的权限操作列表。

状态码： 400

表 12-63 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/datasource/configurations
```

## 响应示例

状态码： 200

OK

```
{  
  "configurations" : [ {  
    "datasource_type": "HIVE",  
    "permission_types": [ "ALLOW" ],  
    "permission_actions": [ "ALL", "SELECT", "UPDATE" ]  
  }  
]
```

```
    }]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ListSecurityDatasourceConfigurationsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListSecurityDatasourceConfigurationsRequest request = new  
        ListSecurityDatasourceConfigurationsRequest();  
        try {  
            ListSecurityDatasourceConfigurationsResponse response =  
            client.listSecurityDatasourceConfigurations(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
import os
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecurityDatasourceConfigurationsRequest()
        response = client.list_security_datasource_configurations(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityDatasourceConfigurationsRequest{}
    response, err := client.ListSecurityDatasourceConfigurations(request)
    if err == nil {
```

```

    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.14 查询权限集中配置的 url 信息

### 功能介绍

查询权限集中配置的url信息，例如obs、hdfs等组件中的url路径。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/permission-sets/datasource/urls

表 12-64 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 12-65 Query 参数

参数	是否必选	参数类型	描述
cluster_id	否	String	集群ID，获取方法请参见 <a href="#">查询单个数据连接信息</a> 。
datasource_type	否	String	数据源类型 <ul style="list-style-type: none"> <li>HIVE数据源</li> </ul>

参数	是否必选	参数类型	描述
parent_permission_set_id	否	String	父权限集ID。获取方法请参见 <a href="#">查询权限集列表</a> 注意： <ul style="list-style-type: none"> <li>当该值为父权限集ID时，则基于父权限集中的权限查询。</li> </ul>

## 请求参数

表 12-66 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

## 响应参数

状态码： 200

表 12-67 响应 Body 参数

参数	参数类型	描述
total	Integer	总条数。
urls	Array of <a href="#">UrlDTO</a> objects	url列表。

表 12-68 UrlDTO

参数	参数类型	描述
name	String	url路径名称。
contains	Boolean	父权限集是否包含此权限, true包含, false不包含, null未检测。



**状态码： 400**

**表 12-69 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/datasource/urls?
cluster_id=dc425074-26b3-479c-9e2f-
b103c0cdd90f&datasource_type=HIVE&parent_permission_set_id=78dea561c983c1157b4ef8c8c746f35f
{
  "cluster_id" : "dc425074-26b3-479c-9e2f-b103c0cdd90f",
  "datasource_type" : "HIVE",
  "parent_permission_set_id" : "78dea561c983c1157b4ef8c8c746f35f"
}
```

## 响应示例

**状态码： 200**

OK

```
{
  "total" : 1,
  "urls" : [{
    "contains" : true,
    "name" : "obs:/root/tmp/test"
  }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityDatasourceUrlsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    }
}
```

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListSecurityDatasourceUrlsRequest request = new ListSecurityDatasourceUrlsRequest();
try {
    ListSecurityDatasourceUrlsResponse response = client.listSecurityDatasourceUrls(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecurityDatasourceUrlsRequest()
        response = client.list_security_datasource_urls(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityDatasourceUrlsRequest{}
    response, err := client.ListSecurityDatasourceUrls(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.1.15 查询数据操作信息

### 功能介绍

查询数据操作信息。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/security/permission-sets/datasource/actions

表 12-70 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 12-71 Query 参数

参数	是否必选	参数类型	描述
parent_permission_set_id	是	String	父权限集ID。获取方法请参见 <a href="#">查询权限集列表</a> 注意： <ul style="list-style-type: none"> <li>当该值为0时，则所有库表均支持查询。</li> <li>当该值为父权限集ID时，则基于父权限集中的权限查询。</li> </ul>
cluster_id	是	String	集群ID，获取方法请参见 <a href="#">查询单个数据连接信息</a> <ul style="list-style-type: none"> <li>查询Hive和DWS数据源操作信息时该数值为必填项，当数据源为DLI时无需填写。</li> </ul>
datasource_type	是	String	数据源类型 <ul style="list-style-type: none"> <li>HIVE数据源</li> <li>DWS数据源</li> <li>DLI数据源</li> </ul>
database_name	否	String	数据库名 注意：该值作为查询关键字时，不能与url同时存在，需要指定其一进行查询。
schema_name	否	String	schema名称。
table_name	否	String	数据表名称。
column_name	否	String	数据字段名称。

参数	是否必选	参数类型	描述
url	否	String	url路径名称 <i>注意：该值作为查询关键字时，不能与database_name同时存在，需要指定其一进行查询</i>

## 请求参数

表 12-72 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

## 响应参数

状态码： 200

表 12-73 响应 Body 参数

参数	参数类型	描述
permission_actions	Array of strings	权限操作列表。

状态码： 400

表 12-74 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/datasource/actions?parent_permission_set_id=0&cluster_id=dc425074-26b3-479c-9e2f-b103c0cdd90f&datasource_type=HIVE&database_name=bigdatatest
```

```
{
  "parent_permission_set_id": "0",
  "cluster_id": "dc425074-26b3-479c-9e2f-b103c0cdd90f",
  "datasource_type": "HIVE",
  "database_name": "bigdatatest"
}
```

## 响应示例

状态码: 200

OK

```
{
  "permission_actions": [ "ALL", "SELECT", "UPDATE", "CREATE", "DROP", "ALTER", "INDEX", "READ",
  "WRITE" ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityDatasourceActionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityDatasourceActionsRequest request = new ListSecurityDatasourceActionsRequest();
        try {
            ListSecurityDatasourceActionsResponse response = client.listSecurityDatasourceActions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
e.printStackTrace();
System.out.println(e.getStatusCode());
System.out.println(e.getRequestId());
System.out.println(e.getErrorCode());
System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecurityDatasourceActionsRequest()
        response = client.list_security_datasource_actions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
```

```
WithAk(ak).
WithSk(sk).
WithProjectId(projectId).
Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListSecurityDatasourceActionsRequest{}
response, err := client.ListSecurityDatasourceActions(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	OK
400	Bad Request

## 12.2 识别规则接口

### 12.2.1 查询识别规则列表

#### 功能介绍

查询识别规则列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/security/data-classification/rule



表 12-75 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

表 12-76 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页显示的条目数量。
offset	否	Integer	偏移量，表示从此偏移量开始查询，该值大于等于0。
secrecy_level	否	String	密级。
name	否	String	规则名称。
creator	否	String	规则创建者。
enable	否	Boolean	规则是否开启。
start_time	否	Long	开始日期。
end_time	否	Long	结束日期。
desc	否	Boolean	排序规则。
order_by	否	String	排序字段, createdAt, createdBy, updatedAt, updatedBy, name, description

## 请求参数

表 12-77 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

## 响应参数

状态码： 200

表 12-78 响应 Body 参数

参数	参数类型	描述
content	Array of <a href="#">DataClassificationRuleQueryDTO</a> objects	查询到的所有数据识别规则。
total	Integer	数据识别规则总数。

表 12-79 DataClassificationRuleQueryDTO

参数	参数类型	描述
uuid	String	规则ID。
rule_type	String	规则类型, CUSTOM, BUILTIN
secrecy_level	String	密级名称。
secrecy_level_num	Long	密级层级。
name	String	规则名称。
guid	String	guid。
enable	Boolean	规则是否开启。
method	String	规则方式, REGULAR, NONE, DEFAULT
content_expression	String	内容表达式。
column_expression	String	列表表达式。
commit_expression	String	备注表达式。
project_id	String	项目ID。
description	String	规则描述。
created_by	String	策略创建人。
created_at	Long	策略创建时间。
updated_by	String	策略更新人。
updated_at	Long	策略更新时间。
builtin_rule_id	String	内置规则ID。
category_id	String	分类ID。

参数	参数类型	描述
instance_id	String	实例ID。
match_type	String	匹配类型。

状态码： 400

表 12-80 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule?offset=0&limit=10
```

## 响应示例

状态码： 200

ok

```
{
  "content": [ {
    "builtin_rule_id": null,
    "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
    "column_expression": null,
    "commit_expression": null,
    "content_expression": ".*",
    "created_at": 1698633124002,
    "created_by": "chenxiaoyu",
    "description": "",
    "enable": true,
    "guid": null,
    "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
    "match_type": null,
    "method": "REGULAR",
    "name": "matchRules",
    "project_id": "0833a5737480d53b2f25c010dc1a7b88",
    "rule_type": "CUSTOM",
    "secrecy_level": "asd",
    "secrecy_level_num": 1,
    "updated_at": 1698633124002,
    "updated_by": "chenxiaoyu",
    "uuid": "8a94800e8b753a35018b7e6be6950023"
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityDataClassificationRulesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityDataClassificationRulesRequest request = new ListSecurityDataClassificationRulesRequest();
        try {
            ListSecurityDataClassificationRulesResponse response =
                client.listSecurityDataClassificationRules(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSecurityDataClassificationRulesRequest()
    response = client.list_security_data_classification_rules(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityDataClassificationRulesRequest{}
    response, err := client.ListSecurityDataClassificationRules(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

状态码	描述
200	ok
400	Bad Request

## 12.2.2 创建识别规则

### 功能介绍

创建识别规则。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/data-classification/rule

表 12-81 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。

### 请求参数

表 12-82 请求 Header 参数

参数	是否必选	参数类型	描述
workspace	是	String	工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。
X-Auth-Token	是	String	IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。

表 12-83 请求 Body 参数

参数	是否必选	参数类型	描述
rule_type	是	String	规则类型, CUSTOM, BUILTIN
secrecy_level_id	是	String	密级ID。
name	是	String	规则名称;rule_type为CUSTOM时用户可自定义名称,rule_type为BUILTIN时输入的名称需要和内置模板中的名称一致。
method	否	String	规则方式, REGULAR, NONE, DEFAULT, COMBINE
content_expression	否	String	内容表达式;只有当rule_type为CUSTOM且method为REGULAR时才必需要此参数。
column_expression	否	String	列表表达式;只有当rule_type为CUSTOM且method为REGULAR时才必需要此参数。
commit_expression	否	String	备注表达式;只有当rule_type为CUSTOM且method为REGULAR时才必需要此参数。
combine_expression	否	String	条件表达式
builtin_rule_id	否	String	内置规则id,只有当rule_type为BUILTIN时才需要此参数。
description	否	String	规则描述。
category_id	否	String	分类ID。
enable	否	Boolean	使能状态。

## 响应参数

状态码： 200

表 12-84 响应 Body 参数

参数	参数类型	描述
uuid	String	规则ID。
rule_type	String	规则类型, CUSTOM, BUILTIN
secrecy_level	String	密级名称。

参数	参数类型	描述
secrecy_level_num	Long	密级层级。
name	String	规则名称。
guid	String	guid。
enable	Boolean	规则是否开启。
method	String	规则方式, REGULAR, NONE, DEFAULT
content_expression	String	内容表达式。
column_expression	String	列表达式。
commit_expression	String	备注表达式。
project_id	String	项目ID。
description	String	规则描述。
created_by	String	策略创建人。
created_at	Long	策略创建时间。
updated_by	String	策略更新人。
updated_at	Long	策略更新时间。
builtin_rule_id	String	内置规则ID。
category_id	String	分类ID。
instance_id	String	实例ID。
match_type	String	匹配类型。

状态码： 400

表 12-85 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误描述。

## 请求示例

- `/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule`



```
{
  "rule_type": "CUSTOM",
  "secrecy_level_id": "8a9480ad7e992660017e9a381244000d",
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "name": "Special_Administrative_Region_test",
  "method": "REGULAR",
  "description": "testModifyDes",
  "content_expression": "testExpression"
}
```

- /v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule

```
{
  "builtin_rule_id": "a5adcca681f011e9a870765a9db3b202",
  "rule_type": "BUILTIN",
  "secrecy_level_id": "8a9480ad7e992660017e9a381244000d",
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "name": "Bank_Account",
  "description": "testCreateBuiltinRule"
}
```

## 响应示例

状态码: 200

ok

```
{
  "builtin_rule_id": null,
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "column_expression": null,
  "commit_expression": null,
  "content_expression": ".*",
  "created_at": 1698633124002,
  "created_by": "chenxiaoyu",
  "description": "",
  "enable": true,
  "guid": null,
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "match_type": null,
  "method": "REGULAR",
  "name": "matchRules",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "rule_type": "CUSTOM",
  "secrecy_level": "asd",
  "secrecy_level_num": 1,
  "updated_at": 1698633124002,
  "updated_by": "chenxiaoyu",
  "uuid": "8a94800e8b753a35018b7e6be6950023"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- ```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class CreateSecurityDataClassificationRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateSecurityDataClassificationRuleRequest request = new
        CreateSecurityDataClassificationRuleRequest();
        DataClassificationRuleOperateDTO body = new DataClassificationRuleOperateDTO();
        body.withCategoryId("0cce38e7c28547828905ae9e4f10a4bf");
        body.withDescription("testModifyDes");
        body.withContentExpression("testExpression");
        body.withMethod(DataClassificationRuleOperateDTO.MethodEnum.fromValue("REGULAR"));
        body.withName("Special_Administrative_Region_test");
        body.withSecrecyLevelId("8a9480ad7e992660017e9a381244000d");
        body.withRuleType(DataClassificationRuleOperateDTO.RuleTypeEnum.fromValue("CUSTOM"));
        request.withBody(body);
        try {
            CreateSecurityDataClassificationRuleResponse response =
            client.createSecurityDataClassificationRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- package com.huaweicloud.sdk.test;

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;
```

```
public class CreateSecurityDataClassificationRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
CreateSecurityDataClassificationRuleRequest request = new
CreateSecurityDataClassificationRuleRequest();
DataClassificationRuleOperateDTO body = new DataClassificationRuleOperateDTO();
body.withCategoryId("0cce38e7c28547828905ae9e4f10a4bf");
body.withDescription("testCreateBuiltinRule");
body.withBuiltinRuleId("a5adcca681f011e9a870765a9db3b202");
body.withName("Bank_Account");
body.withSecrecyLevelId("8a9480ad7e992660017e9a381244000d");
body.withRuleType(DataClassificationRuleOperateDTO.RuleTypeEnum.fromValue("BUILTIN"));
request.withBody(body);
try {
    CreateSecurityDataClassificationRuleResponse response =
client.createSecurityDataClassificationRule(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

- # coding: utf-8  

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = CreateSecurityDataClassificationRuleRequest()  
    request.body = DataClassificationRuleOperateDTO(  
        category_id="0cce38e7c28547828905ae9e4f10a4bf",  
        description="testModifyDes",  
        content_expression="testExpression",  
        method="REGULAR",  
        name="Special_Administrative_Region_test",  
        secrecy_level_id="8a9480ad7e992660017e9a381244000d",  
        rule_type="CUSTOM"  
    )  
    response = client.create_security_data_classification_rule(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

- # coding: utf-8

```
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    environment variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before  
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
    environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateSecurityDataClassificationRuleRequest()  
        request.body = DataClassificationRuleOperateDTO(  
            category_id="0cce38e7c28547828905ae9e4f10a4bf",  
            description="testCreateBuiltInRule",  
            builtin_rule_id="a5adcca681f011e9a870765a9db3b202",  
            name="Bank_Account",  
            secrecy_level_id="8a9480ad7e992660017e9a381244000d",  
            rule_type="BUILTIN"  
        )  
        response = client.create_security_data_classification_rule(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

- ```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityDataClassificationRuleRequest{
        categoryIdDataClassificationRuleOperateDto:= "0cce38e7c28547828905ae9e4f10a4bf"
        descriptionDataClassificationRuleOperateDto:= "testModifyDes"
        contentExpressionDataClassificationRuleOperateDto:= "testExpression"
        methodDataClassificationRuleOperateDto:=
model.GetDataClassificationRuleOperateDtoMethodEnum().REGULAR
        nameDataClassificationRuleOperateDto:= "Special_Administrative_Region_test"
        secrecyLevelIdDataClassificationRuleOperateDto:= "8a9480ad7e992660017e9a381244000d"
        ruleTypeDataClassificationRuleOperateDto:=
model.GetDataClassificationRuleOperateDtoRuleTypeEnum().CUSTOM
        request.Body = &model.DataClassificationRuleOperateDto{
            CategoryId: &categoryIdDataClassificationRuleOperateDto,
            Description: &descriptionDataClassificationRuleOperateDto,
            ContentExpression: &contentExpressionDataClassificationRuleOperateDto,
            Method: &methodDataClassificationRuleOperateDto,
            Name: &nameDataClassificationRuleOperateDto,
            SecrecyLevelId: &secrecyLevelIdDataClassificationRuleOperateDto,
            RuleType: &ruleTypeDataClassificationRuleOperateDto,
        }
        response, err := client.CreateSecurityDataClassificationRule(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}


```
- ```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"

```

```

)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityDataClassificationRuleRequest{
        categoryIdDataClassificationRuleOperateDto:= "0cce38e7c28547828905ae9e4f10a4bf"
        descriptionDataClassificationRuleOperateDto:= "testCreateBuiltinRule"
        builtinRuleIdDataClassificationRuleOperateDto:= "a5adcca681f011e9a870765a9db3b202"
        nameDataClassificationRuleOperateDto:= "Bank_Account"
        secrecyLevelIdDataClassificationRuleOperateDto:= "8a9480ad7e992660017e9a381244000d"
        ruleTypeDataClassificationRuleOperateDto:=
    model.GetDataClassificationRuleOperateDtoRuleTypeEnum().BUILTIN
    request.Body = &model.DataClassificationRuleOperateDto{
        CategoryId: &categoryIdDataClassificationRuleOperateDto,
        Description: &descriptionDataClassificationRuleOperateDto,
        BuiltinRuleId: &builtinRuleIdDataClassificationRuleOperateDto,
        Name: &nameDataClassificationRuleOperateDto,
        SecrecyLevelId: &secrecyLevelIdDataClassificationRuleOperateDto,
        RuleType: &ruleTypeDataClassificationRuleOperateDto,
    }
    response, err := client.CreateSecurityDataClassificationRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.2.3 查询特定识别规则

### 功能介绍

查询特定识别规则。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/data-classification/rule/{id}

表 12-86 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 需要查询的规则ID。                               |

### 请求参数

表 12-87 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

### 响应参数

状态码： 200

表 12-88 响应 Body 参数

| 参数        | 参数类型   | 描述                    |
|-----------|--------|-----------------------|
| uuid      | String | 规则ID。                 |
| rule_type | String | 规则类型, CUSTOM, BUILTIN |

| 参数                 | 参数类型    | 描述                           |
|--------------------|---------|------------------------------|
| secrecy_level      | String  | 密级名称。                        |
| secrecy_level_num  | Long    | 密级层级。                        |
| name               | String  | 规则名称。                        |
| guid               | String  | guid。                        |
| enable             | Boolean | 规则是否开启。                      |
| method             | String  | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String  | 内容表达式。                       |
| column_expression  | String  | 列表表达式。                       |
| commit_expression  | String  | 备注表达式。                       |
| project_id         | String  | 项目ID。                        |
| description        | String  | 规则描述。                        |
| created_by         | String  | 策略创建人。                       |
| created_at         | Long    | 策略创建时间。                      |
| updated_by         | String  | 策略更新人。                       |
| updated_at         | Long    | 策略更新时间。                      |
| builtin_rule_id    | String  | 内置规则ID。                      |
| category_id        | String  | 分类ID。                        |
| instance_id        | String  | 实例ID。                        |
| match_type         | String  | 匹配类型。                        |

状态码： 400

表 12-89 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |



## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/  
8a94800e8b753a35018b7e6be6950023
```

## 响应示例

状态码： 200

ok

```
{  
  "builtin_rule_id" : null,  
  "category_id" : "0cce38e7c28547828905ae9e4f10a4bf",  
  "column_expression" : null,  
  "commit_expression" : null,  
  "content_expression" : ".*",  
  "created_at" : 1698633124002,  
  "created_by" : "chenxiaoyu",  
  "description" : "",  
  "enable" : true,  
  "guid" : null,  
  "instance_id" : "dd97167b873d4a79b2aad54d4370a3bc",  
  "match_type" : null,  
  "method" : "REGULAR",  
  "name" : "matchRules",  
  "project_id" : "0833a5737480d53b2f25c010dc1a7b88",  
  "rule_type" : "CUSTOM",  
  "secrecy_level" : "asd",  
  "secrecy_level_num" : 1,  
  "updated_at" : 1698633124002,  
  "updated_by" : "chenxiaoyu",  
  "uuid" : "8a94800e8b753a35018b7e6be6950023"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowSecurityDataClassificationRuleSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)
```

```
        .withAk(ak)
        .withSk(sk);

    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
        .withCredential(auth)
        .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowSecurityDataClassificationRuleRequest request = new
ShowSecurityDataClassificationRuleRequest();
    request.withId("{id}");
    try {
        ShowSecurityDataClassificationRuleResponse response =
client.showSecurityDataClassificationRule(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecurityDataClassificationRuleRequest()
        request.id = "{id}"
        response = client.show_security_data_classification_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSecurityDataClassificationRuleRequest{}
    request.Id = "{id}"
    response, err := client.ShowSecurityDataClassificationRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.2.4 删除识别规则

### 功能介绍

删除识别规则。

## 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v1/{project\_id}/security/data-classification/rule/{id}

表 12-90 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 需要删除的规则id。                               |

## 请求参数

表 12-91 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 400

表 12-92 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/8a94800e8b753a35018b7e6be6950023
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteSecurityDataClassificationRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteSecurityDataClassificationRuleRequest request = new
        DeleteSecurityDataClassificationRuleRequest();
        request.withId("{id}");
        try {
            DeleteSecurityDataClassificationRuleResponse response =
            client.deleteSecurityDataClassificationRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecurityDataClassificationRuleRequest()
        request.id = "{id}"
        response = client.delete_security_data_classification_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSecurityDataClassificationRuleRequest{}
```

```
request.Id = "{id}"
response, err := client.DeleteSecurityDataClassificationRule(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.2.5 修改识别规则接口

### 功能介绍

修改识别规则接口。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/security/data-classification/rule/{id}

表 12-93 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 识别规则id。                                  |

## 请求参数

表 12-94 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                      |
|--------------|------|--------|-------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token, 通过调用IAM服务获取用户Token接口获取(响应消息头中X-Subject-Token的值)使用Token认证时必选。 |

表 12-95 请求 Body 参数

| 参数                 | 是否必选 | 参数类型    | 描述                                                                   |
|--------------------|------|---------|----------------------------------------------------------------------|
| rule_type          | 是    | String  | 规则类型, CUSTOM, BUILTIN                                                |
| secrecy_level_id   | 是    | String  | 密级ID。                                                                |
| name               | 是    | String  | 规则名称;rule_type为CUSTOM时用户可自定义名称,rule_type为BUILTIN时输入的名称需要和内置模板中的名称一致。 |
| method             | 否    | String  | 规则方式, REGULAR, NONE, DEFAULT, COMBINE                                |
| content_expression | 否    | String  | 内容表达式;只有当rule_type为CUSTOM且method为REGULAR时才必需要此参数。                    |
| column_expression  | 否    | String  | 列表表达式;只有当rule_type为CUSTOM且method为REGULAR时才必需要此参数。                    |
| commit_expression  | 否    | String  | 备注表达式;只有当rule_type为CUSTOM且method为REGULAR时才必需要此参数。                    |
| combine_expression | 否    | String  | 条件表达式                                                                |
| builtin_rule_id    | 否    | String  | 内置规则id,只有当rule_type为BUILTIN时才需要此参数。                                  |
| description        | 否    | String  | 规则描述。                                                                |
| category_id        | 否    | String  | 分类ID。                                                                |
| enable             | 否    | Boolean | 使能状态。                                                                |



## 响应参数

状态码： 200

表 12-96 响应 Body 参数

| 参数                 | 参数类型    | 描述                           |
|--------------------|---------|------------------------------|
| uuid               | String  | 规则ID。                        |
| rule_type          | String  | 规则类型, CUSTOM, BUILTIN        |
| secrecy_level      | String  | 密级名称。                        |
| secrecy_level_num  | Long    | 密级层级。                        |
| name               | String  | 规则名称。                        |
| guid               | String  | guid。                        |
| enable             | Boolean | 规则是否开启。                      |
| method             | String  | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String  | 内容表达式。                       |
| column_expression  | String  | 列表达式。                        |
| commit_expression  | String  | 备注表达式。                       |
| project_id         | String  | 项目ID。                        |
| description        | String  | 规则描述。                        |
| created_by         | String  | 策略创建人。                       |
| created_at         | Long    | 策略创建时间。                      |
| updated_by         | String  | 策略更新人。                       |
| updated_at         | Long    | 策略更新时间。                      |
| builtin_rule_id    | String  | 内置规则ID。                      |
| category_id        | String  | 分类ID。                        |
| instance_id        | String  | 实例ID。                        |
| match_type         | String  | 匹配类型。                        |

状态码： 400

表 12-97 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/8a9480a58b0d64a1018b137553670006
{
  "rule_type": "CUSTOM",
  "secrecy_level_id": "8a9480ad7e992660017e9a381244000d",
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "name": "matchRules",
  "method": "REGULAR",
  "description": "testModifyDesc",
  "content_expression": "testExpression"
}
```

## 响应示例

状态码： 200

ok

```
{
  "builtin_rule_id": null,
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "column_expression": null,
  "commit_expression": null,
  "content_expression": ".*",
  "created_at": 1698633124002,
  "created_by": "chenxiaoyu",
  "description": "",
  "enable": true,
  "guid": null,
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "match_type": null,
  "method": "REGULAR",
  "name": "matchRules",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "rule_type": "CUSTOM",
  "secrecy_level": "asd",
  "secrecy_level_num": 1,
  "updated_at": 1698633124002,
  "updated_by": "chenxiaoyu",
  "uuid": "8a94800e8b753a35018b7e6be6950023"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateSecurityDataClassificationRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSecurityDataClassificationRuleRequest request = new
        UpdateSecurityDataClassificationRuleRequest();
        request.withId("{id}");
        DataClassificationRuleOperateDTO body = new DataClassificationRuleOperateDTO();
        body.withCategoryId("0cce38e7c28547828905ae9e4f10a4bf");
        body.withDescription("testModifyDesc");
        body.withContentExpression("testExpression");
        body.withMethod(DataClassificationRuleOperateDTO.MethodEnum.fromValue("REGULAR"));
        body.withName("matchRules");
        body.withSecrecyLevelId("8a9480ad7e992660017e9a381244000d");
        body.withRuleType(DataClassificationRuleOperateDTO.RuleTypeEnum.fromValue("CUSTOM"));
        request.withBody(body);
        try {
            UpdateSecurityDataClassificationRuleResponse response =
            client.updateSecurityDataClassificationRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecurityDataClassificationRuleRequest()
        request.id = "{id}"
        request.body = DataClassificationRuleOperateDTO(
            category_id="0cce38e7c28547828905ae9e4f10a4bf",
            description="testModifyDesc",
            content_expression="testExpression",
            method="REGULAR",
            name="matchRules",
            secrecy_level_id="8a9480ad7e992660017e9a381244000d",
            rule_type="CUSTOM"
        )
        response = client.update_security_data_classification_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```

Build()

request := &model.UpdateSecurityDataClassificationRuleRequest{}
request.Id = "{id}"
categoryIdDataClassificationRuleOperateDto:= "0cce38e7c28547828905ae9e4f10a4bf"
descriptionDataClassificationRuleOperateDto:= "testModifyDesc"
contentExpressionDataClassificationRuleOperateDto:= "testExpression"
methodDataClassificationRuleOperateDto:=
model.GetDataClassificationRuleOperateDtoMethodEnum().REGULAR
nameDataClassificationRuleOperateDto:= "matchRules"
secrecyLevelIdDataClassificationRuleOperateDto:= "8a9480ad7e992660017e9a381244000d"
ruleTypeIdDataClassificationRuleOperateDto:=
model.GetDataClassificationRuleOperateDtoRuleTypeEnum().CUSTOM
request.Body = &model.DataClassificationRuleOperateDto{
    CategoryId: &categoryIdDataClassificationRuleOperateDto,
    Description: &descriptionDataClassificationRuleOperateDto,
    ContentExpression: &contentExpressionDataClassificationRuleOperateDto,
    Method: &methodDataClassificationRuleOperateDto,
    Name: &nameDataClassificationRuleOperateDto,
    SecrecyLevelId: &secrecyLevelIdDataClassificationRuleOperateDto,
    RuleType: &ruleTypeIdDataClassificationRuleOperateDto,
}
response, err := client.UpdateSecurityDataClassificationRule(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.2.6 批量删除识别规则接口

### 功能介绍

批量删除识别规则接口。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/data-classification/rule/batch-delete

表 12-98 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-99 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-100 请求 Body 参数

| 参数       | 是否必选 | 参数类型             | 描述        |
|----------|------|------------------|-----------|
| rule_ids | 是    | Array of strings | 识别规则id列表。 |

## 响应参数

状态码： 400

表 12-101 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/batch-delete
{
  "rule_ids": [ "d9ff4b06db43e3d4f81de1e60077a480" ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteSecurityDataClassificationRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteSecurityDataClassificationRuleRequest request = new
        BatchDeleteSecurityDataClassificationRuleRequest();
        BatchDeleteRulesBaseDTO body = new BatchDeleteRulesBaseDTO();
        List<String> listbodyRuleIds = new ArrayList<>();
        listbodyRuleIds.add("d9ff4b06db43e3d4f81de1e60077a480");
        body.withRuleIds(listbodyRuleIds);
        request.withBody(body);
        try {
            BatchDeleteSecurityDataClassificationRuleResponse response =
            client.batchDeleteSecurityDataClassificationRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

## Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = BatchDeleteSecurityDataClassificationRuleRequest()  
        listRuleIdsbody = [  
            "d9ff4b06db43e3d4f81de1e60077a480"  
        ]  
        request.body = BatchDeleteRulesBaseDTO(  
            rule_ids=listRuleIdsbody  
        )  
        response = client.batch_delete_security_data_classification_rule(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().
```



```

WithAk(ak).
WithSk(sk).
WithProjectId(projectId).
Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchDeleteSecurityDataClassificationRuleRequest{}
var listRuleIdsbody = []string{
    "d9ff4b06db43e3d4f81de1e60077a480",
}
request.Body = &model.BatchDeleteRulesBaseDto{
    RuleIds: listRuleIdsbody,
}
response, err := client.BatchDeleteSecurityDataClassificationRule(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.2.7 修改识别规则状态接口

### 功能介绍

修改识别规则状态接口。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/security/data-classification/rule/switch-status/{id}

表 12-102 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 识别规则id。                                  |

## 请求参数

表 12-103 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-104 请求 Body 参数

| 参数     | 是否必选 | 参数类型    | 描述        |
|--------|------|---------|-----------|
| enable | 否    | Boolean | 是否开启识别规则。 |

## 响应参数

状态码： 200

表 12-105 响应 Body 参数

| 参数        | 参数类型    | 描述        |
|-----------|---------|-----------|
| rule_id   | String  | 识别规则id。   |
| rule_name | String  | 识别规则名称。   |
| enabled   | Boolean | 识别规则是否开启。 |

状态码： 400

表 12-106 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/switch-status/8a9480a58b0d64a1018b137553670006
{
  "enable" : true
}
```

## 响应示例

状态码： 200

```
ok
{
  "enabled" : true,
  "rule_id" : "8a9480a58b0d64a1018b137553670006",
  "rule_name" : "Special_Administrative_Region"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateSecurityRuleEnableStatusSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateSecurityRuleEnableStatusRequest request = new UpdateSecurityRuleEnableStatusRequest();
request.withId("{id}");
DataClassificationRuleEnableDTO body = new DataClassificationRuleEnableDTO();
body.withEnable(true);
request.withBody(body);
try {
    UpdateSecurityRuleEnableStatusResponse response =
client.updateSecurityRuleEnableStatus(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecurityRuleEnableStatusRequest()
        request.id = "{id}"
        request.body = DataClassificationRuleEnableDTO(
            enable=True
        )
        response = client.update_security_rule_enable_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecurityRuleEnableStatusRequest{}
    request.Id = "{id}"
    enableDataClassificationRuleEnableDto:= true
    request.Body = &model.DataClassificationRuleEnableDto{
        Enable: &enableDataClassificationRuleEnableDto,
    }
    response, err := client.UpdateSecurityRuleEnableStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.2.8 创建组合识别规则

### 功能介绍

创建组合识别规则

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/data-classification/rule/combine

表 12-107 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-108 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-109 请求 Body 参数

| 参数                 | 是否必选 | 参数类型   | 描述    |
|--------------------|------|--------|-------|
| name               | 是    | String | 规则名称  |
| secrecy_level_id   | 是    | String | 密级ID  |
| combine_expression | 是    | String | 条件表达式 |
| description        | 否    | String | 规则描述  |
| category_id        | 否    | String | 分类ID  |

| 参数                 | 是否必选 | 参数类型                                                             | 描述    |
|--------------------|------|------------------------------------------------------------------|-------|
| enable             | 否    | Boolean                                                          | 使能状态。 |
| single_expressions | 是    | Array of <a href="#">DataClassificationSingleRuleDTO</a> objects | 条件单规则 |

表 12-110 DataClassificationSingleRuleDTO

| 参数              | 是否必选 | 参数类型   | 描述                                                                                   |
|-----------------|------|--------|--------------------------------------------------------------------------------------|
| rule_code       | 是    | String | 规则序号,大写字母                                                                            |
| algorithm_type  | 是    | String | 算法类型, REGEX, REGEX_INSENSITIVE, GROOVY, LENGTH_EQ, LENGTH_G<br>T, LENGTH_LT, BUILTIN |
| match_type      | 是    | String | 匹配类型, CONTENT, COLUMN, COMMIT, T<br>ABLE_NAME, TABLE_COMMENT, DATABASE_NAME          |
| expression      | 是    | String | expression                                                                           |
| builtin_rule_id | 否    | String | 内置规则ID                                                                               |

## 响应参数

状态码： 200

表 12-111 响应 Body 参数

| 参数                | 参数类型    | 描述                    |
|-------------------|---------|-----------------------|
| uuid              | String  | 规则ID。                 |
| rule_type         | String  | 规则类型, CUSTOM, BUILTIN |
| secrecy_level     | String  | 密级名称。                 |
| secrecy_level_num | Long    | 密级层级。                 |
| name              | String  | 规则名称。                 |
| guid              | String  | guid。                 |
| enable            | Boolean | 规则是否开启。               |

| 参数                 | 参数类型   | 描述                           |
|--------------------|--------|------------------------------|
| method             | String | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String | 内容表达式。                       |
| column_expression  | String | 列表表达式。                       |
| commit_expression  | String | 备注表达式。                       |
| project_id         | String | 项目ID。                        |
| description        | String | 规则描述。                        |
| created_by         | String | 策略创建人。                       |
| created_at         | Long   | 策略创建时间。                      |
| updated_by         | String | 策略更新人。                       |
| updated_at         | Long   | 策略更新时间。                      |
| builtin_rule_id    | String | 内置规则ID。                      |
| category_id        | String | 分类ID。                        |
| instance_id        | String | 实例ID。                        |
| match_type         | String | 匹配类型。                        |

状态码： 400

表 12-112 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule

```
{
  "secrecy_level_id": "8a9480ad7e992660017e9a381244000d",
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "name": "Special_Administrative_Region_test",
  "description": "testModifyDes",
  "combine_expression": "A&&B",
  "single_expressions": [ {
    "rule_code": "A",
    "algorithm_type": "REGEX",
    "match_type": "CONTENT",
```



```
"expression": "a*"
}, {
  "rule_code": "B",
  "algorithm_type": "LENGTH_EQ",
  "match_type": "TABLE_COMMENT",
  "expression": "11
}]
}
```

## 响应示例

状态码: 200

ok

```
{
  "builtin_rule_id": null,
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "column_expression": null,
  "commit_expression": null,
  "content_expression": "*",
  "created_at": 1698633124002,
  "created_by": "chenxiaoyu",
  "description": "",
  "enable": true,
  "groups": null,
  "guid": null,
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "match_type": null,
  "method": "REGULAR",
  "name": "matchRules",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "rule_type": "CUSTOM",
  "secrecy_level": "asd",
  "secrecy_level_num": 1,
  "updated_at": 1698633124002,
  "updated_by": "chenxiaoyu",
  "uuid": "8a94800e8b753a35018b7e6be6950023"
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.2.9 修改组合识别规则

### 功能介绍

修改组合识别规则

### 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1/{project\_id}/security/data-classification/rule/combine/{id}

表 12-113 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 识别规则id                                   |

## 请求参数

表 12-114 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-115 请求 Body 参数

| 参数                 | 是否必选 | 参数类型                                                             | 描述    |
|--------------------|------|------------------------------------------------------------------|-------|
| name               | 是    | String                                                           | 规则名称  |
| secrecy_level_id   | 是    | String                                                           | 密级ID  |
| combine_expression | 是    | String                                                           | 条件表达式 |
| description        | 否    | String                                                           | 规则描述  |
| category_id        | 否    | String                                                           | 分类ID  |
| enable             | 否    | Boolean                                                          | 使能状态。 |
| single_expressions | 是    | Array of <a href="#">DataClassificationSingleRuleDTO</a> objects | 条件单规则 |

表 12-116 DataClassificationSingleRuleDTO

| 参数              | 是否必选 | 参数类型   | 描述                                                                           |
|-----------------|------|--------|------------------------------------------------------------------------------|
| rule_code       | 是    | String | 规则序号,大写字母                                                                    |
| algorithm_type  | 是    | String | 算法类型, REGEX,REGEX_INSENSITIVE,GR OOVY,LENGTH_EQ,LENGTH_G T,LENGTH_LT,BUILTIN |
| match_type      | 是    | String | 匹配类型, CONTENT,COLUMN,COMMIT,T ABLE_NAME,TABLE_COMMEN T,DATABASE_NAME         |
| expression      | 是    | String | expression                                                                   |
| builtin_rule_id | 否    | String | 内置规则ID                                                                       |

## 响应参数

状态码： 200

表 12-117 响应 Body 参数

| 参数                 | 参数类型    | 描述                           |
|--------------------|---------|------------------------------|
| uuid               | String  | 规则ID。                        |
| rule_type          | String  | 规则类型, CUSTOM, BUILTIN        |
| secrecy_level      | String  | 密级名称。                        |
| secrecy_level_num  | Long    | 密级层级。                        |
| name               | String  | 规则名称。                        |
| guid               | String  | guid。                        |
| enable             | Boolean | 规则是否开启。                      |
| method             | String  | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String  | 内容表达式。                       |
| column_expression  | String  | 列表表达式。                       |
| commit_expression  | String  | 备注表达式。                       |
| project_id         | String  | 项目ID。                        |

| 参数              | 参数类型   | 描述      |
|-----------------|--------|---------|
| description     | String | 规则描述。   |
| created_by      | String | 策略创建人。  |
| created_at      | Long   | 策略创建时间。 |
| updated_by      | String | 策略更新人。  |
| updated_at      | Long   | 策略更新时间。 |
| builtin_rule_id | String | 内置规则ID。 |
| category_id     | String | 分类ID。   |
| instance_id     | String | 实例ID。   |
| match_type      | String | 匹配类型。   |

状态码： 400

表 12-118 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/{id}

{
  "secrecy_level_id": "8a9480ad7e992660017e9a381244000d",
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "name": "Special_Administrative_Region_test",
  "description": "testModifyDes",
  "combine_expression": "A&&B",
  "single_expressions": [ {
    "rule_code": "A",
    "algorithm_type": "REGEX",
    "match_type": "CONTENT",
    "expression": "a*"
  }, {
    "rule_code": "B",
    "algorithm_type": "LENGTH_EQ",
    "match_type": "TABLE_COMMENT",
    "expression": "11"
  } ]
}
```

## 响应示例

状态码： 200

ok

```
{
  "builtin_rule_id": null,
  "category_id": "0cce38e7c28547828905ae9e4f10a4bf",
  "column_expression": null,
  "commit_expression": null,
  "content_expression": ".*",
  "created_at": 1698633124002,
  "created_by": "chenxiaoyu",
  "description": "",
  "enable": true,
  "groups": null,
  "guid": null,
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "method": "REGULAR",
  "name": "matchRules",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "rule_type": "CUSTOM",
  "secrecy_level": "asd",
  "secrecy_level_num": 1,
  "updated_at": 1698633124002,
  "updated_by": "chenxiaoyu",
  "uuid": "8a94800e8b753a35018b7e6be6950023"
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.2.10 组合识别规则测试

### 功能介绍

组合识别规则测试

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/data-classification/rule/check

表 12-119 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-120 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-121 请求 Body 参数

| 参数                     | 是否必选 | 参数类型                                                             | 描述      |
|------------------------|------|------------------------------------------------------------------|---------|
| expression             | 否    | String                                                           | 条件表达式   |
| combine_input_data     | 否    | <a href="#">combine_input_data</a> object                        | 模拟数据    |
| combine                | 否    | Boolean                                                          | 分类ID    |
| single_rule_check_list | 否    | Array of <a href="#">DataClassificationSingleRuleDTO</a> objects | 条件单规则列表 |

表 12-122 combine\_input\_data

| 参数             | 是否必选 | 参数类型   | 描述       |
|----------------|------|--------|----------|
| column_content | 否    | String | 字段内容模拟数据 |
| column_name    | 否    | String | 列名模拟数据   |
| column_comment | 否    | String | 列注释模拟数据  |
| table_comment  | 否    | String | 表注释模拟数据  |
| table_name     | 否    | String | 表名模拟数据   |
| database_name  | 否    | String | 库名模拟数据   |

表 12-123 DataClassificationSingleRuleDTO

| 参数              | 是否必选 | 参数类型   | 描述                                                                           |
|-----------------|------|--------|------------------------------------------------------------------------------|
| rule_code       | 是    | String | 规则序号,大写字母                                                                    |
| algorithm_type  | 是    | String | 算法类型, REGEX,REGEX_INSENSITIVE,GR OOVY,LENGTH_EQ,LENGTH_G T,LENGTH_LT,BUILTIN |
| match_type      | 是    | String | 匹配类型, CONTENT,COLUMN,COMMIT,T ABLE_NAME,TABLE_COMMENT, DATABASE_NAME         |
| expression      | 是    | String | expression                                                                   |
| builtin_rule_id | 否    | String | 内置规则ID                                                                       |

## 响应参数

状态码： 400

表 12-124 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/{id}

```
{
  "expression": "A&&B",
  "combine": true,
  "combine_input_data": {
    "column_content": "1367777xxxx",
    "table_comment": "TEL",
    "column_name": "MOL_TEL"
  },
  "single_rule_check_list": [ {
    "rule_code": "A",
    "algorithm_type": "REGEX",
    "match_type": "CONTENT",
    "expression": "a*"
  }, {
    "rule_code": "B",
    "algorithm_type": "LENGTH_EQ",
    "match_type": "TABLE_COMMENT",
    "expression": "11"
  } ]
}
```

## 响应示例

无

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.3 规则分组接口

### 12.3.1 查询规则组列表

#### 功能介绍

查询规则组列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/security/data-classification/rule/group

表 12-125 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-126 Query 参数

| 参数      | 是否必选 | 参数类型    | 描述                       |
|---------|------|---------|--------------------------|
| limit   | 否    | Integer | 每页显示的条目数量。               |
| offset  | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。 |
| name    | 否    | String  | 规则组名称。                   |
| creator | 否    | String  | 规则组创建者。                  |
| desc    | 否    | Boolean | 排序规则。                    |



| 参数       | 是否必选 | 参数类型   | 描述                                                                  |
|----------|------|--------|---------------------------------------------------------------------|
| order_by | 否    | String | 排序字段, createdAt, createdBy, updatedAt, updatedBy, name, description |

## 请求参数

表 12-127 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                      |
|--------------|------|--------|-------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token, 通过调用IAM服务获取用户Token接口获取(响应消息头中X-Subject-Token的值)使用Token认证时必选。 |

## 响应参数

状态码: 200

表 12-128 响应 Body 参数

| 参数          | 参数类型                                                             | 描述     |
|-------------|------------------------------------------------------------------|--------|
| rule_groups | Array of <a href="#">DataClassificationGroupQueryDTO</a> objects | 规则组列表。 |
| total       | Long                                                             | 规则组总数。 |

表 12-129 DataClassificationGroupQueryDTO

| 参数   | 参数类型   | 描述     |
|------|--------|--------|
| uuid | String | 规则组ID。 |
| name | String | 规则组名称。 |

| 参数          | 参数类型                                                            | 描述       |
|-------------|-----------------------------------------------------------------|----------|
| rules       | Array of <a href="#">DataClassificationRuleQueryDTO</a> objects | 规则实体。    |
| description | String                                                          | 规则组描述。   |
| created_by  | String                                                          | 规则组创建人。  |
| created_at  | Long                                                            | 规则组创建时间。 |
| updated_by  | String                                                          | 规则组更新人。  |
| updated_at  | Long                                                            | 规则组更新时间。 |
| project_id  | String                                                          | 项目ID。    |

表 12-130 DataClassificationRuleQueryDTO

| 参数                 | 参数类型    | 描述                           |
|--------------------|---------|------------------------------|
| uuid               | String  | 规则ID。                        |
| rule_type          | String  | 规则类型, CUSTOM, BUILTIN        |
| secrecy_level      | String  | 密级名称。                        |
| secrecy_level_num  | Long    | 密级层级。                        |
| name               | String  | 规则名称。                        |
| guid               | String  | guid。                        |
| enable             | Boolean | 规则是否开启。                      |
| method             | String  | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String  | 内容表达式。                       |
| column_expression  | String  | 列表表达式。                       |
| commit_expression  | String  | 备注表达式。                       |
| project_id         | String  | 项目ID。                        |
| description        | String  | 规则描述。                        |
| created_by         | String  | 策略创建人。                       |
| created_at         | Long    | 策略创建时间。                      |

| 参数              | 参数类型   | 描述      |
|-----------------|--------|---------|
| updated_by      | String | 策略更新人。  |
| updated_at      | Long   | 策略更新时间。 |
| builtin_rule_id | String | 内置规则ID。 |
| category_id     | String | 分类ID。   |
| instance_id     | String | 实例ID。   |
| match_type      | String | 匹配类型。   |

状态码： 400

表 12-131 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/group?offset=0&limit=10&name=test_rule_groups
```

## 响应示例

状态码： 200

OK

```
{
  "rule_groups": [ {
    "created_at": 1697600262611,
    "created_by": "ei_dayu_test_01",
    "description": "test",
    "name": "test_rule_group",
    "project_id": "0833a5737480d512fa51w010dc1a7b88",
    "rules": null,
    "updated_at": 1697600262611,
    "updated_by": "ei_dayu_test_01",
    "uuid": "8a125fdsesa0b421012saed35c770004"
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityDataClassificationRuleGroupsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityDataClassificationRuleGroupsRequest request = new
        ListSecurityDataClassificationRuleGroupsRequest();
        try {
            ListSecurityDataClassificationRuleGroupsResponse response =
            client.listSecurityDataClassificationRuleGroups(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
```

```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSecurityDataClassificationRuleGroupsRequest()
    response = client.list_security_data_classification_rule_groups(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityDataClassificationRuleGroupsRequest{}
    response, err := client.ListSecurityDataClassificationRuleGroups(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.3.2 创建规则分组接口

### 功能介绍

创建规则分组接口

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/data-classification/rule/group

表 12-132 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-133 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-134 请求 Body 参数

| 参数           | 是否必选 | 参数类型                                                          | 描述      |
|--------------|------|---------------------------------------------------------------|---------|
| name         | 是    | String                                                        | 规则名称    |
| rule_ids     | 是    | Array of strings                                              | 规则id列表  |
| description  | 否    | String                                                        | 规则组描述   |
| create_rules | 否    | Array of <b>DataClassificationGroupCombineRuleDTO</b> objects | 需要创建的规则 |

表 12-135 DataClassificationGroupCombineRuleDTO

| 参数                 | 是否必选 | 参数类型                                                    | 描述            |
|--------------------|------|---------------------------------------------------------|---------------|
| name               | 是    | String                                                  | 规则名称          |
| secrecy_level_id   | 是    | String                                                  | 密级ID          |
| combine_expression | 是    | String                                                  | 条件表达式         |
| description        | 否    | String                                                  | 规则描述          |
| category_id        | 否    | String                                                  | 分类ID          |
| enable             | 否    | Boolean                                                 | 使能状态。         |
| method             | 是    | String                                                  | 规则方式, COMBINE |
| single_expressions | 是    | Array of <b>DataClassificationSingleRuleDTO</b> objects | 条件单规则         |

表 12-136 DataClassificationSingleRuleDTO

| 参数             | 是否必选 | 参数类型   | 描述                                                                           |
|----------------|------|--------|------------------------------------------------------------------------------|
| rule_code      | 是    | String | 规则序号,大写字母                                                                    |
| algorithm_type | 是    | String | 算法类型, REGEX,REGEX_INSENSITIVE,GR OOVY,LENGTH_EQ,LENGTH_G T,LENGTH_LT,BUILTIN |

| 参数              | 是否必选 | 参数类型   | 描述                                                                 |
|-----------------|------|--------|--------------------------------------------------------------------|
| match_type      | 是    | String | 匹配类型, CONTENT,COLUMN,COMMIT,TABLE_NAME,TABLE_COMMENT,DATABASE_NAME |
| expression      | 是    | String | expression                                                         |
| builtin_rule_id | 否    | String | 内置规则ID                                                             |

## 响应参数

状态码： 200

表 12-137 响应 Body 参数

| 参数          | 参数类型                                                   | 描述       |
|-------------|--------------------------------------------------------|----------|
| uuid        | String                                                 | 规则组ID。   |
| name        | String                                                 | 规则组名称。   |
| rules       | Array of <b>DataClassificationRuleQueryDTO</b> objects | 规则实体。    |
| description | String                                                 | 规则组描述。   |
| created_by  | String                                                 | 规则组创建人。  |
| created_at  | Long                                                   | 规则组创建时间。 |
| updated_by  | String                                                 | 规则组更新人。  |
| updated_at  | Long                                                   | 规则组更新时间。 |
| project_id  | String                                                 | 项目ID。    |

表 12-138 DataClassificationRuleQueryDTO

| 参数                | 参数类型   | 描述                    |
|-------------------|--------|-----------------------|
| uuid              | String | 规则ID。                 |
| rule_type         | String | 规则类型, CUSTOM, BUILTIN |
| secrecy_level     | String | 密级名称。                 |
| secrecy_level_num | Long   | 密级层级。                 |



| 参数                 | 参数类型    | 描述                           |
|--------------------|---------|------------------------------|
| name               | String  | 规则名称。                        |
| guid               | String  | guid。                        |
| enable             | Boolean | 规则是否开启。                      |
| method             | String  | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String  | 内容表达式。                       |
| column_expression  | String  | 列表表达式。                       |
| commit_expression  | String  | 备注表达式。                       |
| project_id         | String  | 项目ID。                        |
| description        | String  | 规则描述。                        |
| created_by         | String  | 策略创建人。                       |
| created_at         | Long    | 策略创建时间。                      |
| updated_by         | String  | 策略更新人。                       |
| updated_at         | Long    | 策略更新时间。                      |
| builtin_rule_id    | String  | 内置规则ID。                      |
| category_id        | String  | 分类ID。                        |
| instance_id        | String  | 实例ID。                        |
| match_type         | String  | 匹配类型。                        |

状态码： 400

表 12-139 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/group
```

```
{
  "name": "Special_Administrative_Region_test",
  "description": "testModifyDes",
```

```
"rule_ids" : [ "250f1e5f6cd3d48fd4d10dbd0f94f533" ]  
}
```

## 响应示例

状态码： 200

ok

```
{  
  "created_at" : 1697600262611,  
  "created_by" : "ei_dayu_test_01",  
  "description" : "test",  
  "name" : "test_rule_group",  
  "project_id" : "0833a5737480d512fa51w010dc1a7b88",  
  "rules" : null,  
  "updated_at" : 1697600262611,  
  "updated_by" : "ei_dayu_test_01",  
  "uuid" : "8a125fdsesa0b421012saed35c770004"  
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.3.3 查询规则组

### 功能介绍

查询规则组。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/data-classification/rule/group/{id}

表 12-140 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 需要查询的规则组ID。                              |

## 请求参数

表 12-141 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-142 响应 Body 参数

| 参数          | 参数类型                                                            | 描述       |
|-------------|-----------------------------------------------------------------|----------|
| uuid        | String                                                          | 规则组ID。   |
| name        | String                                                          | 规则组名称。   |
| rules       | Array of <a href="#">DataClassificationRuleQueryDTO</a> objects | 规则实体。    |
| description | String                                                          | 规则组描述。   |
| created_by  | String                                                          | 规则组创建人。  |
| created_at  | Long                                                            | 规则组创建时间。 |
| updated_by  | String                                                          | 规则组更新人。  |
| updated_at  | Long                                                            | 规则组更新时间。 |
| project_id  | String                                                          | 项目ID。    |

表 12-143 DataClassificationRuleQueryDTO

| 参数            | 参数类型   | 描述                    |
|---------------|--------|-----------------------|
| uuid          | String | 规则ID。                 |
| rule_type     | String | 规则类型, CUSTOM, BUILTIN |
| secrecy_level | String | 密级名称。                 |

| 参数                 | 参数类型    | 描述                           |
|--------------------|---------|------------------------------|
| secrecy_level_num  | Long    | 密级层级。                        |
| name               | String  | 规则名称。                        |
| guid               | String  | guid。                        |
| enable             | Boolean | 规则是否开启。                      |
| method             | String  | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String  | 内容表达式。                       |
| column_expression  | String  | 列表达式。                        |
| commit_expression  | String  | 备注表达式。                       |
| project_id         | String  | 项目ID。                        |
| description        | String  | 规则描述。                        |
| created_by         | String  | 策略创建人。                       |
| created_at         | Long    | 策略创建时间。                      |
| updated_by         | String  | 策略更新人。                       |
| updated_at         | Long    | 策略更新时间。                      |
| builtin_rule_id    | String  | 内置规则ID。                      |
| category_id        | String  | 分类ID。                        |
| instance_id        | String  | 实例ID。                        |
| match_type         | String  | 匹配类型。                        |

状态码： 400

表 12-144 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/group/ped96c90125691b8bf8d6886407b816t
```

## 响应示例

状态码： 200

ok

```
{
  "created_at" : 1697600262611,
  "created_by" : "ei_dayu_test_01",
  "description" : "test",
  "name" : "test_rule_group",
  "project_id" : "0833a5737480d512fa51w010dc1a7b88",
  "rules" : null,
  "updated_at" : 1697600262611,
  "updated_by" : "ei_dayu_test_01",
  "uuid" : "8a125fdsesa0b421012saed35c770004"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowSecurityDataClassificationRuleGroupSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        ShowSecurityDataClassificationRuleGroupRequest request = new
        ShowSecurityDataClassificationRuleGroupRequest();
        request.withId("{id}");
        try {
            ShowSecurityDataClassificationRuleGroupResponse response =
            client.showSecurityDataClassificationRuleGroup(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecurityDataClassificationRuleGroupRequest()
        request.id = "{id}"
        response = client.show_security_data_classification_rule_group(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowSecurityDataClassificationRuleGroupRequest{}
request.Id = "{id}"
response, err := client.ShowSecurityDataClassificationRuleGroup(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.3.4 修改规则分组接口

### 功能介绍

修改规则分组接口

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/security/data-classification/rule/group/{id}

表 12-145 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 需要查询的规则组ID                               |

## 请求参数

表 12-146 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-147 请求 Body 参数

| 参数          | 是否必选 | 参数类型             | 描述     |
|-------------|------|------------------|--------|
| name        | 是    | String           | 规则名称   |
| rule_ids    | 是    | Array of strings | 规则id列表 |
| description | 否    | String           | 规则组描述  |

## 响应参数

状态码： 200

表 12-148 响应 Body 参数

| 参数   | 参数类型   | 描述     |
|------|--------|--------|
| uuid | String | 规则组ID。 |
| name | String | 规则组名称。 |



| 参数          | 参数类型                                                            | 描述       |
|-------------|-----------------------------------------------------------------|----------|
| rules       | Array of <a href="#">DataClassificationRuleQueryDTO</a> objects | 规则实体。    |
| description | String                                                          | 规则组描述。   |
| created_by  | String                                                          | 规则组创建人。  |
| created_at  | Long                                                            | 规则组创建时间。 |
| updated_by  | String                                                          | 规则组更新人。  |
| updated_at  | Long                                                            | 规则组更新时间。 |
| project_id  | String                                                          | 项目ID。    |

表 12-149 DataClassificationRuleQueryDTO

| 参数                 | 参数类型    | 描述                           |
|--------------------|---------|------------------------------|
| uuid               | String  | 规则ID。                        |
| rule_type          | String  | 规则类型, CUSTOM, BUILTIN        |
| secrecy_level      | String  | 密级名称。                        |
| secrecy_level_num  | Long    | 密级层级。                        |
| name               | String  | 规则名称。                        |
| guid               | String  | guid。                        |
| enable             | Boolean | 规则是否开启。                      |
| method             | String  | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String  | 内容表达式。                       |
| column_expression  | String  | 列表表达式。                       |
| commit_expression  | String  | 备注表达式。                       |
| project_id         | String  | 项目ID。                        |
| description        | String  | 规则描述。                        |
| created_by         | String  | 策略创建人。                       |
| created_at         | Long    | 策略创建时间。                      |

| 参数              | 参数类型   | 描述      |
|-----------------|--------|---------|
| updated_by      | String | 策略更新人。  |
| updated_at      | Long   | 策略更新时间。 |
| builtin_rule_id | String | 内置规则ID。 |
| category_id     | String | 分类ID。   |
| instance_id     | String | 实例ID。   |
| match_type      | String | 匹配类型。   |

状态码： 400

表 12-150 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/group/
ped96c90125691b8bf8d6886407b816t
{
  "name": "Special_Administrative_Region_test",
  "description": "testModifyDes",
  "rule_ids": [ "250f1e5f6cd3d48fd4d10dbd0f94f533" ]
}
```

## 响应示例

状态码： 200

ok

```
{
  "created_at": 1697600262611,
  "created_by": "ei_dayu_test_01",
  "description": "test",
  "name": "test_rule_group",
  "project_id": "0833a5737480d512fa51w010dc1a7b88",
  "rules": null,
  "updated_at": 1697600262611,
  "updated_by": "ei_dayu_test_01",
  "uuid": "8a125fdsesa0b421012saed35c770004"
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.3.5 删除规则分组接口

### 功能介绍

删除规则分组接口

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/data-classification/rule/group/batch-delete

表 12-151 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-152 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-153 请求 Body 参数

| 参数             | 是否必选 | 参数类型             | 描述        |
|----------------|------|------------------|-----------|
| rule_group_ids | 是    | Array of strings | 识别规则组id列表 |

## 响应参数

状态码： 400

表 12-154 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/rule/group/batch-delete
{
  "rule_group_ids": [ "d9ff4b06db43e3d4f81de1e60077a480" ]
}
```

## 响应示例

无

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.4 数据权限查询接口

### 12.4.1 查询角色对一组库、表的可配置操作权限信息

#### 功能介绍

查询角色对一组库、表的可配置操作权限信息。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/security/permission-sets/datasource/role/batch-actions

表 12-155 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-156 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-157 请求 Body 参数

| 参数                       | 是否必选 | 参数类型             | 描述                                                               |
|--------------------------|------|------------------|------------------------------------------------------------------|
| parent_permission_set_id | 是    | String           | 父权限集id。                                                          |
| cluster_id               | 是    | String           | 集群id。                                                            |
| datasource_type          | 是    | String           | 数据源类型, HIVE                                                      |
| database_names           | 是    | Array of strings | 目前批量授权只支持单库下的多表授权，或同一集群下多个多库授权，区分这两类可通过 传参中 tables是否为空来判断。       |
| schemas                  | 否    | Array of strings | dws权限涉及 schema，预留字段，在做DWS批量授权时应保持单schema下的批量授权，或者对单库下schema批量授权。 |

| 参数           | 是否必选 | 参数类型             | 描述      |
|--------------|------|------------------|---------|
| table_names  | 否    | Array of strings | 数据表列表。  |
| column_names | 否    | Array of strings | 数据字段列表。 |

## 响应参数

状态码： 200

表 12-158 响应 Body 参数

| 参数                 | 参数类型             | 描述      |
|--------------------|------------------|---------|
| permission_actions | Array of strings | 权限操作列表。 |

状态码： 400

表 12-159 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-sets/datasource/role/batch-actions
{
  "parent_permission_set_id": "24dd53c92ddb12fe762f9d9d42a031a",
  "cluster_id": "dc425074-26b3-479c-9e2f-b103c0cdd90f",
  "datasource_type": "HIVE",
  "database_names": [ "default" ],
  "table_names": [ "auto_test_qwaw" ],
  "column_names": [ ]
}
```

## 响应示例

状态码： 200

OK

```
{
  "permission_actions": [ "ALL", "SELECT", "UPDATE", "CREATE", "DROP", "ALTER", "INDEX", "READ", "WRITE" ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ListSecurityRoleActionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityRoleActionsRequest request = new ListSecurityRoleActionsRequest();
        RoleActionQueryDTO body = new RoleActionQueryDTO();
        List<String> listbodyTableNames = new ArrayList<>();
        listbodyTableNames.add("auto_test_qwaw");
        List<String> listbodyDatabaseNames = new ArrayList<>();
        listbodyDatabaseNames.add("default");
        body.withTableNames(listbodyTableNames);
        body.withDatabaseNames(listbodyDatabaseNames);
        body.withDatasourceType(RoleActionQueryDTO.DatasourceTypeEnum.fromValue("HIVE"));
        body.withClusterId("dc425074-26b3-479c-9e2f-b103c0cdd90f");
        body.withParentPermissionSetId("24dd53c92ddb12fe762f9d9d42a031a");
        request.withBody(body);
        try {
            ListSecurityRoleActionsResponse response = client.listSecurityRoleActions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecurityRoleActionsRequest()
        listTableNamesbody = [
            "auto_test_qwaw"
        ]
        listDatabaseNamesbody = [
            "default"
        ]
        request.body = RoleActionQueryDTO(
            table_names=listTableNamesbody,
            database_names=listDatabaseNamesbody,
            datasource_type="HIVE",
            cluster_id="dc425074-26b3-479c-9e2f-b103c0cdd90f",
            parent_permission_set_id="24dd53c92ddb12fe762f9d9d42a031a"
        )
        response = client.list_security_role_actions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```



```

sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListSecurityRoleActionsRequest{}
var listTableNamesbody = []string{
    "auto_test_qwaw",
}
var listDatabaseNamesbody = []string{
    "default",
}
request.Body = &model.RoleActionQueryDto{
    TableNames: &listTableNamesbody,
    DatabaseNames: listDatabaseNamesbody,
    DatasourceType: model.GetRoleActionQueryDtoDatasourceTypeEnum().HIVE,
    ClusterId: "dc425074-26b3-479c-9e2f-b103c0cdd90f",
    ParentPermissionSetId: "24dd53c92ddb12fe762f9d9d42a031a",
}
response, err := client.ListSecurityRoleActions(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.5 数据密级接口

### 12.5.1 获取数据密级

#### 功能介绍

获取数据密级。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/security/data-classification/secretcy-level

表 12-160 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-161 Query 参数

| 参数       | 是否必选 | 参数类型    | 描述                                                                                                                                                                                                                                     |
|----------|------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| offset   | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。                                                                                                                                                                                                               |
| limit    | 否    | Integer | 每页显示的条目数量。                                                                                                                                                                                                                             |
| order_by | 否    | String  | 排序字段 <ul style="list-style-type: none"> <li>• CREATED_AT 按照创建时间排序</li> <li>• CREATED_BY 按照创建人排序</li> <li>• UPDATED_AT 按照更新时间排序</li> <li>• UPDATED_BY 按照更新人排序</li> <li>• NAME 按照数据密级名称排序</li> <li>• DESCRIPTION 按照数据密级描述信息排序</li> </ul> |
| desc     | 否    | Boolean | 排序规则。                                                                                                                                                                                                                                  |

## 请求参数

表 12-162 请求 Header 参数

| 参数        | 是否必选 | 参数类型   | 描述                                           |
|-----------|------|--------|----------------------------------------------|
| workspace | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。 |

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-163 响应 Body 参数

| 参数      | 参数类型                                          | 描述      |
|---------|-----------------------------------------------|---------|
| total   | Integer                                       | 总条数。    |
| content | Array of <a href="#">SecrecyLevel</a> objects | 数据密级列表。 |

表 12-164 SecrecyLevel

| 参数                   | 参数类型    | 描述            |
|----------------------|---------|---------------|
| secrecy_level_id     | String  | 数据密级id。       |
| secrecy_level_name   | String  | 数据密级名称。       |
| secrecy_level_number | Integer | 数据密级等级。       |
| description          | String  | 数据密级描述。       |
| created_by           | String  | 创建者。          |
| created_at           | Long    | 创建时间。         |
| updated_by           | String  | 更新者。          |
| updated_at           | Long    | 更新时间。         |
| instance_id          | String  | DataArts实例ID。 |

状态码： 400

表 12-165 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/secrecy-level?
order_by=CREATED_AT&desc=false
```

## 响应示例

状态码： 200

OK

```
{
  "content": [ {
    "created_at": 1668998595733,
    "created_by": "ei_dayu_xxxxxx_01",
    "description": "test secrecy level",
    "instance_id": "ff1b509366614b23a51866233c696501",
    "secrecy_level_id": "8a948095841c87e90184980d01a20000",
    "secrecy_level_name": "test_secrecy_level",
    "secrecy_level_number": 1,
    "updated_at": 1709558070704,
    "updated_by": "ei_dayu_xxxxxx_01"
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecuritySecrecyLevelsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListSecuritySecrecyLevelsRequest request = new ListSecuritySecrecyLevelsRequest();
try {
    ListSecuritySecrecyLevelsResponse response = client.listSecuritySecrecyLevels(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

## Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecuritySecrecyLevelsRequest()
        response = client.list_security_secrecy_levels(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecuritySecrecyLevelsRequest{}
    response, err := client.ListSecuritySecrecyLevels(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.5.2 创建数据密级

### 功能介绍

创建数据密级。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/security/data-classification/secrecy-level

表 12-166 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-167 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-168 请求 Body 参数

| 参数          | 是否必选 | 参数类型   | 描述                                 |
|-------------|------|--------|------------------------------------|
| name        | 否    | String | 数据密级名称。名称应该唯一，只能由英文字母、数字、下划线、汉字构成。 |
| description | 否    | String | 数据密级描述。                            |

## 响应参数

状态码： 200

表 12-169 响应 Body 参数

| 参数                   | 参数类型    | 描述            |
|----------------------|---------|---------------|
| secrecy_level_id     | String  | 数据密级id。       |
| secrecy_level_name   | String  | 数据密级名称。       |
| secrecy_level_number | Integer | 数据密级等级。       |
| description          | String  | 数据密级描述。       |
| created_by           | String  | 创建者。          |
| created_at           | Long    | 创建时间。         |
| updated_by           | String  | 更新者。          |
| updated_at           | Long    | 更新时间。         |
| instance_id          | String  | DataArts实例ID。 |

状态码： 400

表 12-170 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/secrecy-level
{
  "name": "test_secrecy_level",
  "description": "test secrecy level"
}
```

## 响应示例

状态码： 200

OK

```
{
  "created_at": 1668998595733,
  "created_by": "ei_dayu_xxxxxx_01",
  "description": "test secrecy level",
  "instance_id": "ff1b509366614b23a51866233c696501",
  "secrecy_level_id": "8a948095841c87e90184980d01a20000",
  "secrecy_level_name": "test_secrecy_level",
```



```
"secrecy_level_number" : 1,  
"updated_at" : 1709558070704,  
"updated_by" : "ei_dayu_xxxxxx_01"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class CreateSecuritySecrecyLevelSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateSecuritySecrecyLevelRequest request = new CreateSecuritySecrecyLevelRequest();  
        CreateSecrecyLevelDTO body = new CreateSecrecyLevelDTO();  
        body.withDescription("test secrecy level");  
        body.withName("test_secrecy_level");  
        request.withBody(body);  
        try {  
            CreateSecuritySecrecyLevelResponse response = client.createSecuritySecrecyLevel(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecuritySecrecyLevelRequest()
        request.body = CreateSecrecyLevelDTO(
            description="test secrecy level",
            name="test_secrecy_level"
        )
        response = client.create_security_secrecy_level(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateSecuritySecrecyLevelRequest{}
descriptionCreateSecrecyLevelDto:= "test secrecy level"
nameCreateSecrecyLevelDto:= "test_secrecy_level"
request.Body = &model.CreateSecrecyLevelDto{
    Description: &descriptionCreateSecrecyLevelDto,
    Name: &nameCreateSecrecyLevelDto,
}
response, err := client.CreateSecuritySecrecyLevel(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.5.3 根据指定的 id 查询数据密级

### 功能介绍

根据指定的id查询数据密级。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/data-classification/secrecy-level/{id}

表 12-171 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

| 参数 | 是否必选 | 参数类型   | 描述      |
|----|------|--------|---------|
| id | 是    | String | 数据密级id。 |

## 请求参数

表 12-172 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-173 响应 Body 参数

| 参数                   | 参数类型    | 描述            |
|----------------------|---------|---------------|
| secrecy_level_id     | String  | 数据密级id。       |
| secrecy_level_name   | String  | 数据密级名称。       |
| secrecy_level_number | Integer | 数据密级等级。       |
| description          | String  | 数据密级描述。       |
| created_by           | String  | 创建者。          |
| created_at           | Long    | 创建时间。         |
| updated_by           | String  | 更新者。          |
| updated_at           | Long    | 更新时间。         |
| instance_id          | String  | DataArts实例ID。 |

状态码： 400

表 12-174 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/secrecy-level/8a9480a78d210ae2018d3413428c0001
```

## 响应示例

状态码： 200

OK

```
{
  "created_at" : 1668998595733,
  "created_by" : "ei_dayu_xxxxxx_01",
  "description" : "test secrecy level",
  "instance_id" : "ff1b509366614b23a51866233c696501",
  "secrecy_level_id" : "8a948095841c87e90184980d01a20000",
  "secrecy_level_name" : "test_secrecy_level",
  "secrecy_level_number" : 1,
  "updated_at" : 1709558070704,
  "updated_by" : "ei_dayu_xxxxxx_01"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowSecuritySecrecyLevelSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
```

```

        .withAk(ak)
        .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ShowSecuritySecrecyLevelRequest request = new ShowSecuritySecrecyLevelRequest();
request.withId("{id}");
try {
    ShowSecuritySecrecyLevelResponse response = client.showSecuritySecrecyLevel(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

## Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecuritySecrecyLevelRequest()
        request.id = "{id}"
        response = client.show_security_secrecy_level(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

## Go

```

package main

import (

```

```

"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSecuritySecrecyLevelRequest{}
    request.Id = "{id}"
    response, err := client.ShowSecuritySecrecyLevel(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.5.4 删除指定的 id 的数据密级

### 功能介绍

删除指定的id的数据密级。

## 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v1/{project\_id}/security/data-classification/secrecy-level/{id}

表 12-175 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 数据密级id。                                  |

## 请求参数

表 12-176 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 400

表 12-177 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/secrecy-level/8a9480a78d210ae2018d3413428c0001
```



## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteSecuritySecrecyLevelSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteSecuritySecrecyLevelRequest request = new DeleteSecuritySecrecyLevelRequest();
        request.withId("{id}");
        try {
            DeleteSecuritySecrecyLevelResponse response = client.deleteSecuritySecrecyLevel(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecuritySecrecyLevelRequest()
        request.id = "{id}"
        response = client.delete_security_secrecy_level(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSecuritySecrecyLevelRequest{}
    request.Id = "{id}"
```

```
response, err := client.DeleteSecuritySecrecyLevel(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.5.5 根据指定的 id 修改数据密级

### 功能介绍

根据指定的id修改数据密级。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/security/data-classification/secrecy-level/{id}

表 12-178 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 数据密级id。                                  |

## 请求参数

表 12-179 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-180 请求 Body 参数

| 参数          | 是否必选 | 参数类型   | 描述      |
|-------------|------|--------|---------|
| description | 否    | String | 数据密级描述。 |

## 响应参数

状态码： 200

表 12-181 响应 Body 参数

| 参数                   | 参数类型    | 描述            |
|----------------------|---------|---------------|
| secrecy_level_id     | String  | 数据密级id。       |
| secrecy_level_name   | String  | 数据密级名称。       |
| secrecy_level_number | Integer | 数据密级等级。       |
| description          | String  | 数据密级描述。       |
| created_by           | String  | 创建者。          |
| created_at           | Long    | 创建时间。         |
| updated_by           | String  | 更新者。          |
| updated_at           | Long    | 更新时间。         |
| instance_id          | String  | DataArts实例ID。 |

状态码： 400

表 12-182 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/secrecy-level/8a9480a78d210ae2018d3413428c0001
{
  "description": "test modify the secrecy level"
}
```

## 响应示例

状态码： 200

OK

```
{
  "created_at": 1668998595733,
  "created_by": "ei_dayu_xxxxxx_01",
  "description": "test modify the secrecy level",
  "instance_id": "ff1b509366614b23a51866233c696501",
  "secrecy_level_id": "8a948095841c87e90184980d01a20000",
  "secrecy_level_name": "test_secrecy_level",
  "secrecy_level_number": 1,
  "updated_at": 1709558070704,
  "updated_by": "ei_dayu_xxxxxx_01"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateSecuritySecrecyLevelSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateSecuritySecrecyLevelRequest request = new UpdateSecuritySecrecyLevelRequest();
request.withId("{id}");
UpdateSecrecyLevelDTO body = new UpdateSecrecyLevelDTO();
body.withDescription("test modify the secrecy level");
request.withBody(body);
try {
    UpdateSecuritySecrecyLevelResponse response = client.updateSecuritySecrecyLevel(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecuritySecrecyLevelRequest()
        request.id = "{id}"
        request.body = UpdateSecrecyLevelDTO(
            description="test modify the secrecy level"
        )
        response = client.update_security_secrecy_level(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecuritySecrecyLevelRequest{}
    request.Id = "{id}"
    descriptionUpdateSecrecyLevelDto := "test modify the secrecy level"
    request.Body = &model.UpdateSecrecyLevelDto{
        Description: &descriptionUpdateSecrecyLevelDto,
    }
    response, err := client.UpdateSecuritySecrecyLevel(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.5.6 批量删除数据密级

### 功能介绍

批量删除数据密级。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/data-classification/secrecy-level/batch-delete

表 12-183 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-184 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-185 请求 Body 参数

| 参数  | 是否必选 | 参数类型             | 描述                         |
|-----|------|------------------|----------------------------|
| ids | 否    | Array of strings | 数据密级id列表，数据密级id可以通过查询接口获取。 |

### 响应参数

状态码： 400



表 12-186 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-classification/secrecy-level/batch-delete
{
  "ids" : [ 1705975693964 ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteSecuritySecrecyLevelsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteSecuritySecrecyLevelsRequest request = new BatchDeleteSecuritySecrecyLevelsRequest();
        BatchDeleteSecrecyLevelDTO body = new BatchDeleteSecrecyLevelDTO();
```

```
List<String> listbodyIds = new ArrayList<>();
listbodyIds.add("1705975693964");
body.withIds(listbodyIds);
request.withBody(body);
try {
    BatchDeleteSecuritySecrecyLevelsResponse response =
client.batchDeleteSecuritySecrecyLevels(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteSecuritySecrecyLevelsRequest()
        listbody = [
            "1705975693964"
        ]
        request.body = BatchDeleteSecrecyLevelDTO(
            ids=listbody
        )
        response = client.batch_delete_security_secrecy_levels(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteSecuritySecrecyLevelsRequest{}
    var listIdsbody = []string{
        "1705975693964",
    }
    request.Body = &model.BatchDeleteSecrecyLevelDto{
        Ids: &listIdsbody,
    }
    response, err := client.BatchDeleteSecuritySecrecyLevels(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.6 权限应用接口

## 12.6.1 查询数据开发细粒度连接列表

### 功能介绍

查询数据开发细粒度连接列表（全量）。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/fgac/dlf/datawarehouses

表 12-187 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-188 Query 参数

| 参数     | 是否必选 | 参数类型    | 描述                       |
|--------|------|---------|--------------------------|
| limit  | 否    | Integer | 每页显示的条目数量。               |
| offset | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。 |

### 请求参数

表 12-189 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

### 响应参数

状态码： 200

表 12-190 响应 Body 参数

| 参数       | 参数类型                                     | 描述           |
|----------|------------------------------------------|--------------|
| dw_lists | Array of <b>DataWarehouseDTO</b> objects | 数据开发细粒度连接列表。 |

表 12-191 DataWarehouseDTO

| 参数        | 参数类型                    | 描述                                                                                                        |
|-----------|-------------------------|-----------------------------------------------------------------------------------------------------------|
| dw_id     | String                  | 数据连接ID。                                                                                                   |
| dw_name   | String                  | 数据连接名称。                                                                                                   |
| dw_type   | String                  | 数据连接类型： <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• SPARK数据源</li> </ul> |
| dw_config | <b>dw_config</b> object | 数据开发细粒度连接配置信息。                                                                                            |

表 12-192 dw\_config

| 参数                   | 参数类型    | 描述                                                                                                                                                                    |
|----------------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fgac_flag            | Boolean | 是否开启细粒度认证,true表示开启细粒度认证,false表示关闭细粒度认证。                                                                                                                               |
| fgac_type            | String  | 细粒度认证类型,开启细粒度认证时才生效。"0"表示开发态细粒度认证,支持数据开发细粒度脚本运行、作业测试运行,"1"表示调度态细粒度认证,支持数据开发细粒度脚本运行、作业测试运行、作业执行调度。                                                                     |
| fgac_conn_status     | String  | 数据源连通性测试状态： <ul style="list-style-type: none"> <li>• UNKNOWN - 连通性未测试</li> <li>• TESTING - 连通性测试中</li> <li>• SUCCESS - 连通性测试成功</li> <li>• FAILED - 连通性测试失败</li> </ul> |
| fgac_conn_test_time  | Long    | 最近一次连通性测试时间。                                                                                                                                                          |
| fgac_conn_test_error | String  | 连通性测试失败信息,如果连通性测试成功或者未测试连通性,失败信息为空字符串。                                                                                                                                |

状态码： 400

表 12-193 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/fgac/dlf/datawarehouses
```

## 响应示例

状态码： 200

OK

```
{
  "dw_lists": [ {
    "dw_config": {
      "fgac_conn_status": "SUCCESS",
      "fgac_conn_test_error": "",
      "fgac_conn_test_time": 1709175082675,
      "fgac_flag": false,
      "fgac_type": "0"
    },
    "dw_id": "7c8a2d85d917492bb3195377cd9c36be",
    "dw_name": "hive",
    "dw_type": "HIVE"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityDlfDataWareHousesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
```

```
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
ListSecurityDlfDataWareHousesRequest request = new ListSecurityDlfDataWareHousesRequest();
try {
    ListSecurityDlfDataWareHousesResponse response = client.listSecurityDlfDataWareHouses(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecurityDlfDataWareHousesRequest()
        response = client.list_security_dlf_data_ware_houses(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityDlfDataWareHousesRequest{}
    response, err := client.ListSecurityDlfDataWareHouses(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.6.2 批量更新数据开发连接细粒度认证状态

### 功能介绍

批量更新数据开发连接细粒度认证状态。



## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1/{project\_id}/security/fgac/df/datawarehouses

表 12-194 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-195 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-196 请求 Body 参数

| 参数       | 是否必选 | 参数类型                                                 | 描述             |
|----------|------|------------------------------------------------------|----------------|
| fgac_ids | 否    | Array of <a href="#">FgacSingleUpdateReq</a> objects | 细粒度认证数据开发连接列表。 |

表 12-197 FgacSingleUpdateReq

| 参数        | 是否必选 | 参数类型    | 描述                                              |
|-----------|------|---------|-------------------------------------------------|
| dw_id     | 否    | String  | 数据连接id，获取方法请参见 <a href="#">查询细粒度认证数据源连接列表</a> 。 |
| fgac_flag | 否    | Boolean | 是否开启细粒度认证,true表示开启细粒度认证,false表示关闭细粒度认证。         |

| 参数        | 是否必选 | 参数类型   | 描述                                                                                                |
|-----------|------|--------|---------------------------------------------------------------------------------------------------|
| fgac_type | 否    | String | 细粒度认证类型，开启细粒度认证时才生效。"0"表示开发态细粒度认证，支持数据开发细粒度脚本运行、作业测试运行，"1"表示调度态细粒度认证，支持数据开发细粒度脚本运行、作业测试运行、作业执行调度。 |

## 响应参数

状态码： 200

表 12-198 响应 Body 参数

| 参数      | 参数类型                                              | 描述         |
|---------|---------------------------------------------------|------------|
| results | Array of <a href="#">FgacUpdateResult</a> objects | 细粒度认证更新结果。 |

表 12-199 FgacUpdateResult

| 参数        | 参数类型    | 描述                             |
|-----------|---------|--------------------------------|
| id        | String  | 数据连接id。                        |
| status    | Boolean | 是否更新成功,true表示更新成功,false表示更新失败。 |
| error_msg | String  | 细粒度认证更新错误信息。                   |

状态码： 400

表 12-200 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/fgac/dlf/datawarehouses
{
```

```
"fgac_ids" : [ {  
  "dw_id" : "7c8a2d85d917492bb3195377cd9c36be",  
  "fgac_flag" : true,  
  "fgac_type" : "1"  
} ]  
}
```

## 响应示例

**状态码： 200**

OK

```
{  
  "results" : [ {  
    "id" : "7c8a2d85d917492bb3195377cd9c36be",  
    "status" : true,  
    "error_msg" : ""  
  } ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class BatchUpdateSecurityDlfDataWareHousesSolution {  
  
  public static void main(String[] args) {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    // environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running  
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    String ak = System.getenv("CLOUD_SDK_AK");  
    String sk = System.getenv("CLOUD_SDK_SK");  
    String projectId = "{project_id}";  
  
    ICredential auth = new BasicCredentials()  
      .withProjectId(projectId)  
      .withAk(ak)  
      .withSk(sk);  
  
    DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
      .withCredential(auth)  
      .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
      .build();  
  
    BatchUpdateSecurityDlfDataWareHousesRequest request = new  
    BatchUpdateSecurityDlfDataWareHousesRequest();  
    FgacUpdateReq body = new FgacUpdateReq();  
    List<FgacSingleUpdateReq> listbodyFgacIds = new ArrayList<>();  
    listbodyFgacIds.add(  

```

```
        new FgacSingleUpdateReq()
            .withDwid("7c8a2d85d917492bb3195377cd9c36be")
            .withFgacFlag(true)
            .withFgacType("1")
    );
    body.withFgacIds(listbodyFgacIds);
    request.withBody(body);
    try {
        BatchUpdateSecurityDlfDataWareHousesResponse response =
client.batchUpdateSecurityDlfDataWareHouses(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchUpdateSecurityDlfDataWareHousesRequest()
        listFgacIdsbody = [
            FgacSingleUpdateReq(
                dw_id="7c8a2d85d917492bb3195377cd9c36be",
                fgac_flag=True,
                fgac_type="1"
            )
        ]
        request.body = FgacUpdateReq(
            fgac_ids=listFgacIdsbody
        )
        response = client.batch_update_security_dlf_data_ware_houses(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchUpdateSecurityDlfDataWareHousesRequest{}
    dwldFgacIds:= "7c8a2d85d917492bb3195377cd9c36be"
    fgacFlagFgacIds:= true
    fgacTypeFgacIds:= "1"
    var listFgacIdsbody = []model.FgacSingleUpdateReq{
        {
            Dwld: &dwldFgacIds,
            FgacFlag: &fgacFlagFgacIds,
            FgacType: &fgacTypeFgacIds,
        },
    }
    request.Body = &model.FgacUpdateReq{
        FgacIds: &listFgacIdsbody,
    }
    response, err := client.BatchUpdateSecurityDlfDataWareHouses(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.6.3 测试数据开发连接细粒度连通性

### 功能介绍

测试数据开发连接细粒度连通性。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/fgac/dlf/datawarehouses/{dw\_id}

表 12-201 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                              |
|------------|------|--------|-------------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。        |
| dw_id      | 是    | String | 数据连接id，获取方法请参见 <a href="#">查询细粒度认证数据源连接列表</a> 。 |

### 请求参数

表 12-202 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 400

表 12-203 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/fgac/dlf/datawarehouses/7c8a2d85d917492bb3195377cd9c36be
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DebugSecurityDlfDataWareHousesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DebugSecurityDlfDataWareHousesRequest request = new DebugSecurityDlfDataWareHousesRequest();
```

```
request.withDwid("{dw_id}");
try {
    DebugSecurityDlfDataWareHousesResponse response =
client.debugSecurityDlfDataWareHouses(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DebugSecurityDlfDataWareHousesRequest()
        request.dw_id = "{dw_id}"
        response = client.debug_security_dlf_data_ware_houses(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)
```



```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DebugSecurityDlfDataWareHousesRequest{}
    request.DwId = "{dw_id}"
    response, err := client.DebugSecurityDlfDataWareHouses(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No content  |
| 400 | Bad Request |

# 12.7 敏感数据分布接口

## 12.7.1 查询敏感数据发现概览结果(以分类和密级为单位)

### 功能介绍

查询敏感数据发现概览结果(以分类和密级为单位)。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/security/sensitive-data/result/overview

表 12-204 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-205 Query 参数

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                    |
|---------------|------|--------|-------------------------------------------------------------------------------------------------------|
| datasource    | 否    | String | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| cluster_name  | 否    | String | 集群名称。                                                                                                 |
| database_name | 否    | String | 数据库名称。                                                                                                |
| schema_name   | 否    | String | schema名称。                                                                                             |
| table_name    | 否    | String | 表名称。                                                                                                  |

## 请求参数

表 12-206 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-207 响应 Body 参数

| 参数                       | 参数类型                                                                       | 描述         |
|--------------------------|----------------------------------------------------------------------------|------------|
| secrecy_level_statistics | Array of <a href="#">SensitiveDataSecrecyLevelOverviewQueryDTO</a> objects | 基于密级的概览统计。 |
| category_statistics      | Array of <a href="#">SensitiveDataCategoryOverviewQueryDTO</a> objects     | 基于分类的概览统计。 |

表 12-208 SensitiveDataSecrecyLevelOverviewQueryDTO

| 参数                   | 参数类型   | 描述            |
|----------------------|--------|---------------|
| secrecy_level_id     | String | 密级ID。         |
| secrecy_level_name   | String | 密级名称。         |
| secrecy_level_number | Long   | 密级的等级。        |
| count                | Long   | 当前密级下的敏感字段数量。 |

表 12-209 SensitiveDataCategoryOverviewQueryDTO

| 参数            | 参数类型   | 描述            |
|---------------|--------|---------------|
| root_id       | String | 当前分类节点的根节点id。 |
| parent_id     | String | 当前分类节点的父节点id。 |
| category_id   | String | 分类的Id。        |
| category_name | String | 分类的名称。        |
| category_path | String | 分类path。       |
| count         | Long   | 当前分类下的敏感字段数量。 |

| 参数       | 参数类型                                                          | 描述        |
|----------|---------------------------------------------------------------|-----------|
| children | Array of <b>SensitiveDataCategoryOverviewQueryDTO</b> objects | 当前分类的子节点。 |

状态码： 400

表 12-210 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

/v1/0833a5737480d53b2f25c010dc1a7b88/security/sensitive-data/result/overview

## 响应示例

状态码： 200

ok

```
{
  "category_statistics": [ {
    "category_id": "07500201b1734a459896b80a8fd7fa59",
    "category_name": "test1",
    "category_path": "test1",
    "children": [ {
      "category_id": "028c98f833d14109906a4a2a27a87547",
      "category_name": "test5",
      "category_path": "test1/test5",
      "children": [ ],
      "count": 0,
      "parent_id": "07500201b1734a459896b80a8fd7fa59",
      "root_id": "07500201b1734a459896b80a8fd7fa59"
    } ],
    "category_id": "13e9eeb0d0ed4de88c1df3262f9b333c",
    "category_name": "test3",
    "category_path": "test1/test3",
    "children": [ ],
    "count": 0,
    "parent_id": "07500201b1734a459896b80a8fd7fa59",
    "root_id": "07500201b1734a459896b80a8fd7fa59"
  } ],
  {
    "category_id": "4603eaff332544e4858fa7f328ecea2a",
    "category_name": "test6",
    "category_path": "test1/test6",
    "children": [ ],
    "count": 0,
    "parent_id": "07500201b1734a459896b80a8fd7fa59",
    "root_id": "07500201b1734a459896b80a8fd7fa59"
  } ],
  {
    "category_id": "07500201b1734a459896b80a8fd7fa59",
    "category_name": "test1",
    "category_path": "test1",
    "children": [ ],
    "count": 0,
    "parent_id": "07500201b1734a459896b80a8fd7fa59",
    "root_id": "07500201b1734a459896b80a8fd7fa59"
  } ]
}
```

```
"category_id" : "5b46ec0ffc9249b59b9bcb4db2a276bf",
"category_name" : "test2",
"category_path" : "test1/test2",
"children" : [ ],
"count" : 0,
"parent_id" : "07500201b1734a459896b80a8fd7fa59",
"root_id" : "07500201b1734a459896b80a8fd7fa59"
}, {
  "category_id" : "b82cb60a51ef47d8a1585cc1f1a75612",
  "category_name" : "test4",
  "category_path" : "test1/test4",
  "children" : [ ],
  "count" : 0,
  "parent_id" : "07500201b1734a459896b80a8fd7fa59",
  "root_id" : "07500201b1734a459896b80a8fd7fa59"
}],
"count" : 4,
"parent_id" : "-1",
"root_id" : "07500201b1734a459896b80a8fd7fa59"
}],
"secrecy_level_statistics" : [ {
  "count" : 6,
  "secrecy_level_id" : "8a948048831b060101831b2ee3be0000",
  "secrecy_level_name" : "testSecrecyLevel01",
  "secrecy_level_number" : 2
} ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecuritySensitiveDataOverviewsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecuritySensitiveDataOverviewsRequest request = new
```

```
ListSecuritySensitiveDataOverviewsRequest();
try {
    ListSecuritySensitiveDataOverviewsResponse response =
client.listSecuritySensitiveDataOverviews(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecuritySensitiveDataOverviewsRequest()
        response = client.list_security_sensitive_data_overviews(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListSecuritySensitiveDataOverviewsRequest{}
response, err := client.ListSecuritySensitiveDataOverviews(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.8 用户同步接口

### 12.8.1 查询用户同步列表

#### 功能介绍

查询用户同步列表。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/security/member-sync-tasks

表 12-211 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                                         |
|------------|------|--------|------------------------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID</a> 和 <a href="#">账号ID</a> 。 |

表 12-212 Query 参数

| 参数              | 是否必选 | 参数类型    | 描述                                                                                                                                                                           |
|-----------------|------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| limit           | 否    | Integer | 每页显示的条目数量。                                                                                                                                                                   |
| offset          | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。                                                                                                                                                     |
| cluster_type    | 否    | String  | 集群类型 <ul style="list-style-type: none"> <li>• MRS数据源</li> <li>• DWS数据源</li> </ul>                                                                                            |
| cluster_name    | 否    | String  | 集群名称。                                                                                                                                                                        |
| sync_status     | 否    | String  | 同步状态 <ul style="list-style-type: none"> <li>• UNKNOWN 未知</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> </ul>    |
| schedule_status | 否    | String  | 用户同步任务调度状态 <ul style="list-style-type: none"> <li>• NOT_SCHEDULE 未启用调度</li> <li>• SCHEDULING 调度中</li> </ul>                                                                  |
| order_by        | 否    | String  | 排序字段 <ul style="list-style-type: none"> <li>• CLUSTER_NAME 按照集群名称排序</li> <li>• CREATE_TIME 按照创建时间排序</li> <li>• UPDATE_TIME 按照更新时间排序</li> <li>• SYNC_TIME 按照同步时间排序</li> </ul> |



| 参数           | 是否必选 | 参数类型    | 描述                |
|--------------|------|---------|-------------------|
| order_by_asc | 否    | Boolean | 是否升序（仅指定排序参数时有效）。 |

## 请求参数

表 12-213 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-214 响应 Body 参数

| 参数    | 参数类型                                             | 描述         |
|-------|--------------------------------------------------|------------|
| total | Long                                             | 用户同步任务总条数。 |
| tasks | Array of <a href="#">MemberSync Task</a> objects | 用户同步任务列表。  |

表 12-215 MemberSyncTask

| 参数                        | 参数类型   | 描述          |
|---------------------------|--------|-------------|
| id                        | String | 用户同步任务id。   |
| project_id                | String | 项目ID。       |
| domain_id                 | String | 租户ID。       |
| instance_id               | String | 实例ID。       |
| data_connection_workspace | String | 数据连接工作空间ID。 |

| 参数                   | 参数类型    | 描述                                                                                                                                                                        |
|----------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cluster_type         | String  | 集群类型 <ul style="list-style-type: none"> <li>• MRS集群</li> <li>• DWS集群</li> </ul>                                                                                           |
| data_connection_id   | String  | 数据连接id。                                                                                                                                                                   |
| data_connection_name | String  | 数据连接名称。                                                                                                                                                                   |
| cluster_id           | String  | 集群id。                                                                                                                                                                     |
| cluster_name         | String  | 集群名称。                                                                                                                                                                     |
| schedule_start_hour  | Integer | 调度开始时间, 单位为小时, 0~23。                                                                                                                                                      |
| schedule_end_hour    | Integer | 调度结束时间, 单位为小时, 0~23。                                                                                                                                                      |
| schedule_period      | String  | 调度周期 <ul style="list-style-type: none"> <li>• MINUTE 分钟为单位调度</li> <li>• HOUR 小时为单位调度</li> </ul>                                                                           |
| schedule_interval    | Integer | 调度间隔。                                                                                                                                                                     |
| schedule_status      | String  | 调度状态 <ul style="list-style-type: none"> <li>• NOT_SCHEDULE 未启用任务调度</li> <li>• SCHEDULING 任务调度中</li> </ul>                                                                 |
| sync_status          | String  | 同步状态 <ul style="list-style-type: none"> <li>• UNKNOWN 未知</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> </ul> |
| sync_msg             | String  | 同步日志。                                                                                                                                                                     |
| sync_time            | Long    | 同步时间。                                                                                                                                                                     |
| create_time          | Long    | 创建时间。                                                                                                                                                                     |
| create_user          | String  | 创建者。                                                                                                                                                                      |
| update_time          | Long    | 更新时间。                                                                                                                                                                     |
| update_user          | String  | 更新者。                                                                                                                                                                      |

状态码： 400

表 12-216 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/member-sync-tasks
```

## 响应示例

状态码： 200

OK

```
{
  "tasks": [ {
    "cluster_id": "c94a9133-ef65-47ad-9483-bfa8c5eaa101",
    "cluster_name": "dws_ssl_4autotest_nomodify",
    "cluster_type": "DWS",
    "create_time": 1710856132000,
    "create_user": "ei_dayu_xxxx_01",
    "data_connection_id": "3190d3fa8bad4de19c20572b181eb00c",
    "data_connection_name": "dws_0319",
    "data_connection_workspace": "f7bc8f2a71524d268c4293974e3dc35b",
    "domain_id": "0833a5736980d53b0f22c0102ffcbfc0",
    "id": "088c2b59114405ab4fbc0df5bd05202b",
    "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
    "project_id": "0833a5737480d53b2f25c010dc1a7b88",
    "schedule_end_hour": 6,
    "schedule_interval": 4,
    "schedule_period": "HOUR",
    "schedule_start_hour": 0,
    "schedule_status": "NOT_SCHEDULE",
    "sync_msg": "[2024-03-20 10:25:54] ==> Submit job to datamanager failed:[{"duration":0.0,"jobId
\\":\\"9324a84d-e151-44ee-aafc-c6a6c2084b0c\\",\\"jobType\\":\\"\\",\\"message\\":\\"ERROR: role \\
\\iam_group_xxxx\\" cannot be dropped because some objects depend on it\\n Detail: target of row level
security policy dls_xxxx_userANDgroups on table all_data_type_0805_03\\ntarget of row level security policy
dls_xxxx_ALLgroups on table all_data_type_0805_03\\ntarget of row level security policy dls_xxxx_asdasdsd
on table absx\\",\\"rowCount\\":0,\\"rows\\":[],\\"schema\\":[],\\"status\\":\\"FAILED\\",\\"totalRowCount\\":0}],
    "sync_status": "SYNC_FAIL",
    "sync_time": 1710901516000,
    "update_time": 1710856132000,
    "update_user": "ei_dayu_xxxx_01"
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityMemberSyncTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityMemberSyncTasksRequest request = new ListSecurityMemberSyncTasksRequest();
        try {
            ListSecurityMemberSyncTasksResponse response = client.listSecurityMemberSyncTasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSecurityMemberSyncTasksRequest()
    response = client.list_security_member_sync_tasks(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityMemberSyncTasksRequest{}
    response, err := client.ListSecurityMemberSyncTasks(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.8.2 查询单个用户同步任务

### 功能介绍

查询单个用户同步任务。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/member-sync-tasks/{task\_id}

表 12-217 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| task_id    | 是    | String | 用户同步任务id。                                |

### 请求参数

表 12-218 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

### 响应参数

状态码： 200

表 12-219 响应 Body 参数

| 参数                        | 参数类型    | 描述                                                                                                                                                                        |
|---------------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id                        | String  | 用户同步任务id。                                                                                                                                                                 |
| project_id                | String  | 项目ID。                                                                                                                                                                     |
| domain_id                 | String  | 租户ID。                                                                                                                                                                     |
| instance_id               | String  | 实例ID。                                                                                                                                                                     |
| data_connection_workspace | String  | 数据连接工作空间ID。                                                                                                                                                               |
| cluster_type              | String  | 集群类型 <ul style="list-style-type: none"> <li>• MRS集群</li> <li>• DWS集群</li> </ul>                                                                                           |
| data_connection_id        | String  | 数据连接id。                                                                                                                                                                   |
| data_connection_name      | String  | 数据连接名称。                                                                                                                                                                   |
| cluster_id                | String  | 集群id。                                                                                                                                                                     |
| cluster_name              | String  | 集群名称。                                                                                                                                                                     |
| schedule_start_hour       | Integer | 调度开始时间, 单位为小时, 0~23。                                                                                                                                                      |
| schedule_end_hour         | Integer | 调度结束时间, 单位为小时, 0~23。                                                                                                                                                      |
| schedule_period           | String  | 调度周期 <ul style="list-style-type: none"> <li>• MINUTE 分钟为单位调度</li> <li>• HOUR 小时为单位调度</li> </ul>                                                                           |
| schedule_interval         | Integer | 调度间隔。                                                                                                                                                                     |
| schedule_status           | String  | 调度状态 <ul style="list-style-type: none"> <li>• NOT_SCHEDULE 未启用任务调度</li> <li>• SCHEDULING 任务调度中</li> </ul>                                                                 |
| sync_status               | String  | 同步状态 <ul style="list-style-type: none"> <li>• UNKNOWN 未知</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> </ul> |
| sync_msg                  | String  | 同步日志。                                                                                                                                                                     |

| 参数          | 参数类型   | 描述    |
|-------------|--------|-------|
| sync_time   | Long   | 同步时间。 |
| create_time | Long   | 创建时间。 |
| create_user | String | 创建者。  |
| update_time | Long   | 更新时间。 |
| update_user | String | 更新者。  |

状态码： 400

表 12-220 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

无

## 响应示例

状态码： 200

OK

```
{
  "cluster_id": "c94a9133-ef65-47ad-9483-bfa8c5eaa101",
  "cluster_name": "dws_ssl_4autotest_nomodify",
  "cluster_type": "DWS",
  "create_time": 1710856132000,
  "create_user": "ei_dayu_xxxx_01",
  "data_connection_id": "3190d3fa8bad4de19c20572b181eb00c",
  "data_connection_name": "dws_0319",
  "data_connection_workspace": "f7bc8f2a71524d268c4293974e3dc35b",
  "domain_id": "0833a5736980d53b0f22c0102ffcbfc0",
  "id": "088c2b59114405ab4fbc0df5bd05202b",
  "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
  "project_id": "0833a5737480d53b2f25c010dc1a7b88",
  "schedule_end_hour": 6,
  "schedule_interval": 4,
  "schedule_period": "HOUR",
  "schedule_start_hour": 0,
  "schedule_status": "NOT_SCHEDULE",
  "sync_msg": "[2024-03-20 10:25:54] ==> Submit job to datamanager failed:{{\"duration\":0.0,\"jobId\": \"9324a84d-e151-44ee-aafc-c6a6c2084b0c\", \"jobType\": \"\", \"message\": \"ERROR: role \\\"iam_group_xxxx\\\" cannot be dropped because some objects depend on it\\n Detail: target of row level security policy dls_xxxx_userANDgroups on table all_data_type_0805_03\\ntarget of row level security policy dls_xxxx_ALLgroups on table all_data_type_0805_03\\ntarget of row level security policy dls_xxxx_asdasdsd on table absx\", \"rowCount\":0, \"rows\": [], \"schema\": [], \"status\": \"FAILED\", \"totalRowCount\":0}}\",
  "sync_status": "SYNC_FAIL",
  "sync_time": 1710901516000,
```



```
"update_time" : 1710856132000,  
"update_user" : "ei_dayu_xxxx_01"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;  
import com.huaweicloud.sdk.dataartsstudio.v1.*;  
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;  
  
public class ShowSecurityMemberSyncTaskSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowSecurityMemberSyncTaskRequest request = new ShowSecurityMemberSyncTaskRequest();  
        request.withTaskId("{task_id}");  
        try {  
            ShowSecurityMemberSyncTaskResponse response = client.showSecurityMemberSyncTask(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8  
import os
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecurityMemberSyncTaskRequest()
        request.task_id = "{task_id}"
        response = client.show_security_member_sync_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSecurityMemberSyncTaskRequest{}
    request.TaskId = "{task_id}"
```

```
response, err := client.ShowSecurityMemberSyncTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

# 12.9 队列权限接口

## 12.9.1 查询当前空间下分配的队列资源

### 功能介绍

查询当前空间下分配的队列资源。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/permission/queue/assigned-source

表 12-221 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-222 Query 参数

| 参数   | 是否必选 | 参数类型   | 描述            |
|------|------|--------|---------------|
| type | 否    | String | 队列类型，MRS、DLI。 |

| 参数         | 是否必选 | 参数类型    | 描述                                        |
|------------|------|---------|-------------------------------------------|
| cluster_id | 否    | String  | 集群ID，获取方法请参见 <a href="#">查询单个数据连接信息</a> 。 |
| limit      | 否    | Integer | 每页显示的条目数量。                                |
| offset     | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。                  |

## 请求参数

表 12-223 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-224 响应 Body 参数

| 参数            | 参数类型                                  | 描述      |
|---------------|---------------------------------------|---------|
| project_id    | String                                | 项目id。   |
| queue_sources | Array of QueueSrcAssignEntity objects | 队列资源信息。 |

表 12-225 QueueSrcAssignEntity

| 参数          | 参数类型   | 描述        |
|-------------|--------|-----------|
| id          | String | 队列资源id。   |
| source_type | String | 队列资源服务名称。 |
| queue_name  | String | 队列名称。     |

| 参数           | 参数类型    | 描述                                      |
|--------------|---------|-----------------------------------------|
| queue_type   | String  | 队列类型。                                   |
| queue_attr   | Integer | 队列属性(0:默认,1:实时队列,2:离线队列), 当前只有yarn队列涉及。 |
| conn_id      | String  | 数据连接id。                                 |
| conn_name    | String  | 数据连接名称。                                 |
| cluster_id   | String  | 集群id。                                   |
| cluster_name | String  | 集群名称。                                   |
| create_time  | Long    | 队列加入此空间的时间。                             |
| create_user  | String  | 队列加入此的操作人。                              |
| update_time  | Long    | 当前空间下管理的队列更新时间。                         |
| update_user  | String  | 当前空间下管理的队列更新人。                          |
| project_id   | String  | 项目id。                                   |
| description  | String  | 当前空间分配资源附加的描述信息。                        |

状态码： 400

表 12-226 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

无

## 响应示例

状态码： 200

OK

```
{
  "project_id": "0833a5737480d53b2f25c010dc1a7b88-workspace-f7bc8f2a71524d268c4293974e3dc35b",
  "queue_sources": [ {
    "cluster_id": "ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
    "cluster_name": "mrs_noauth_autotest_do_not_del",
    "conn_id": "8e2bd0e9aa24401ea1a8edf83cd50323",
    "conn_name": "ranger_0324_noauth",
    "create_time": 1711424611000,
    "create_user": "ei_dayu_xxxx_01",
```

```
"description" : "",
"id" : "0412bf3bf03f42689c6a02d93ff8f306",
"project_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-f7bc8f2a71524d268c4293974e3dc35b",
"queue_attr" : 0,
"queue_name" : "root.default",
"queue_type" : "yarn",
"source_type" : "mrs",
"update_time" : null,
"update_user" : null
} ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityAssignedQueuesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityAssignedQueuesRequest request = new ListSecurityAssignedQueuesRequest();
        try {
            ListSecurityAssignedQueuesResponse response = client.listSecurityAssignedQueues(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecurityAssignedQueuesRequest()
        response = client.list_security_assigned_queues(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.ListSecurityAssignedQueuesRequest{}  
response, err := client.ListSecurityAssignedQueues(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.9.2 分配队列资源给指定空间

### 功能介绍

分配队列资源给指定空间。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission/queue/assigned-source

表 12-227 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |



## 请求参数

表 12-228 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-229 请求 Body 参数

| 参数          | 是否必选 | 参数类型             | 描述                    |
|-------------|------|------------------|-----------------------|
| source_type | 否    | String           | 队列资源服务(当前只支持mrs、dli)。 |
| queue_name  | 否    | Array of strings | 队列名称。                 |
| conn_id     | 否    | String           | 数据连接id。               |
| cluster_id  | 否    | String           | 集群id。                 |
| description | 否    | String           | 当前空间分配资源附加的描述信息。      |

## 响应参数

状态码： 200

表 12-230 响应 Body 参数

| 参数     | 参数类型                                                 | 描述 |
|--------|------------------------------------------------------|----|
| [数组元素] | Array<Array< <a href="#">QueueSrcAssignEntity</a> >> | ok |

表 12-231 QueueSrcAssignEntity

| 参数 | 参数类型   | 描述      |
|----|--------|---------|
| id | String | 队列资源id。 |

| 参数           | 参数类型    | 描述                                      |
|--------------|---------|-----------------------------------------|
| source_type  | String  | 队列资源服务名称。                               |
| queue_name   | String  | 队列名称。                                   |
| queue_type   | String  | 队列类型。                                   |
| queue_attr   | Integer | 队列属性(0:默认,1:实时队列,2:离线队列), 当前只有yarn队列涉及。 |
| conn_id      | String  | 数据连接id。                                 |
| conn_name    | String  | 数据连接名称。                                 |
| cluster_id   | String  | 集群id。                                   |
| cluster_name | String  | 集群名称。                                   |
| create_time  | Long    | 队列加入此空间的时间。                             |
| create_user  | String  | 队列加入此的操作人。                              |
| update_time  | Long    | 当前空间下管理的队列更新时间。                         |
| update_user  | String  | 当前空间下管理的队列更新人。                          |
| project_id   | String  | 项目id。                                   |
| description  | String  | 当前空间分配资源附加的描述信息。                        |

状态码： 400

表 12-232 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission/queue/assigned-source
{
  "source_type": "mrs",
  "queue_name": [ "root.dgc_test" ],
  "conn_id": "8e2bd0e9aa24401ea1a8edf83cd50323",
  "cluster_id": "ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
  "description": ""
}
```

## 响应示例

状态码： 200

ok

```
[ {
  "cluster_id" : "ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
  "cluster_name" : "mrs_noauth_autotest_do_not_del",
  "conn_id" : "8e2bd0e9aa24401ea1a8edf83cd50323",
  "conn_name" : "ranger_0324_noauth",
  "create_time" : 1712493119159,
  "create_user" : "ei_dayu_xxxx_01",
  "description" : "",
  "id" : "8c92991966c1456b9175f3c93a08ee4e",
  "project_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-f7bc8f2a71524d268c4293974e3dc35b",
  "queue_attr" : 0,
  "queue_name" : "root.dgc_test",
  "queue_type" : "yarn",
  "source_type" : "mrs",
  "update_time" : null,
  "update_user" : null
} ]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateSecurityAssignedQueueSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateSecurityAssignedQueueRequest request = new CreateSecurityAssignedQueueRequest();
        QueueSrcAssignCreateDTO body = new QueueSrcAssignCreateDTO();
        List<String> listbodyQueueName = new ArrayList<>();
        listbodyQueueName.add("root.dgc_test");
        body.withDescription("");
        body.withClusterId("ec26b187-626a-4e89-bd1f-6824fbcfb2b6");
        body.withConnId("8e2bd0e9aa24401ea1a8edf83cd50323");
    }
}
```

```
body.withQueueName(listbodyQueueName);
body.withSourceType("mrs");
request.withBody(body);
try {
    CreateSecurityAssignedQueueResponse response = client.createSecurityAssignedQueue(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecurityAssignedQueueRequest()
        listQueueNamebody = [
            "root.dgc_test"
        ]
        request.body = QueueSrcAssignCreateDTO(
            description="",
            cluster_id="ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
            conn_id="8e2bd0e9aa24401ea1a8edf83cd50323",
            queue_name=listQueueNamebody,
            source_type="mrs"
        )
        response = client.create_security_assigned_queue(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityAssignedQueueRequest{}
    var listQueueNamebody = []string{
        "root.dgc_test",
    }
    descriptionQueueSrcAssignCreateDto := ""
    clusterIdQueueSrcAssignCreateDto := "ec26b187-626a-4e89-bd1f-6824fbcfb2b6"
    connIdQueueSrcAssignCreateDto := "8e2bd0e9aa24401ea1a8edf83cd50323"
    sourceTypeQueueSrcAssignCreateDto := "mrs"
    request.Body = &model.QueueSrcAssignCreateDto{
        Description: &descriptionQueueSrcAssignCreateDto,
        ClusterId: &clusterIdQueueSrcAssignCreateDto,
        ConnId: &connIdQueueSrcAssignCreateDto,
        QueueName: &listQueueNamebody,
        SourceType: &sourceTypeQueueSrcAssignCreateDto,
    }
    response, err := client.CreateSecurityAssignedQueue(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.9.3 修改当前空间下分配的队列资源

### 功能介绍

修改当前空间下分配的队列资源。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/security/permission/queue/assigned-source/{id}

表 12-233 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 分配给当前空间的队列资源id。                          |

### 请求参数

表 12-234 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-235 请求 Body 参数

| 参数          | 是否必选 | 参数类型    | 描述                                      |
|-------------|------|---------|-----------------------------------------|
| queue_attr  | 否    | Integer | 队列属性(0:默认,1:实时队列,2:离线队列), 当前只有yarn队列涉及。 |
| description | 否    | String  | 当前空间分配资源附加的描述信息。                        |

## 响应参数

状态码： 200

表 12-236 响应 Body 参数

| 参数           | 参数类型    | 描述                                      |
|--------------|---------|-----------------------------------------|
| id           | String  | 队列资源id。                                 |
| source_type  | String  | 队列资源服务名称。                               |
| queue_name   | String  | 队列名称。                                   |
| queue_type   | String  | 队列类型。                                   |
| queue_attr   | Integer | 队列属性(0:默认,1:实时队列,2:离线队列), 当前只有yarn队列涉及。 |
| conn_id      | String  | 数据连接id。                                 |
| conn_name    | String  | 数据连接名称。                                 |
| cluster_id   | String  | 集群id。                                   |
| cluster_name | String  | 集群名称。                                   |
| create_time  | Long    | 队列加入此空间的时间。                             |
| create_user  | String  | 队列加入此的操作人。                              |
| update_time  | Long    | 当前空间下管理的队列更新时间。                         |
| update_user  | String  | 当前空间下管理的队列更新人。                          |
| project_id   | String  | 项目id。                                   |
| description  | String  | 当前空间分配资源附加的描述信息。                        |

状态码： 400

表 12-237 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/{project_id}/security/permission/queue/assigned-source/8c92991966c1456b9175f3c93a08ee4e
{
  "queue_attr" : 1
}
```

## 响应示例

状态码： 200

ok

```
{
  "cluster_id" : "ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
  "cluster_name" : "mrs_noauth_autotest_do_not_del",
  "conn_id" : "8e2bd0e9aa24401ea1a8edf83cd50323",
  "conn_name" : "ranger_0324_noauth",
  "create_time" : 1712493119000,
  "create_user" : "ei_dayu_xxxx_01",
  "description" : null,
  "id" : "8c92991966c1456b9175f3c93a08ee4e",
  "project_id" : "0833a5737480d53b2f25c010dc1a7b88-workspace-f7bc8f2a71524d268c4293974e3dc35b",
  "queue_attr" : 1,
  "queue_name" : "root.dgc_test",
  "queue_type" : "yarn",
  "source_type" : "mrs",
  "update_time" : 1712541740529,
  "update_user" : "ei_dayu_xxxx_01"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class UpdateSecurityAssignedQueueSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```



```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
    .withCredential(auth)
    .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateSecurityAssignedQueueRequest request = new UpdateSecurityAssignedQueueRequest();
request.withId("{id}");
QueueSrcAssignUpdateDTO body = new QueueSrcAssignUpdateDTO();
body.withQueueAttr(1);
request.withBody(body);
try {
    UpdateSecurityAssignedQueueResponse response = client.updateSecurityAssignedQueue(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecurityAssignedQueueRequest()
        request.id = "{id}"
        request.body = QueueSrcAssignUpdateDTO(
```

```
        queue_attr=1
    )
    response = client.update_security_assigned_queue(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecurityAssignedQueueRequest{
        request.Id = "{id}"
        queueAttrQueueSrcAssignUpdateDto:= int32(1)
        request.Body = &model.QueueSrcAssignUpdateDto{
            QueueAttr: &queueAttrQueueSrcAssignUpdateDto,
        }
    }
    response, err := client.UpdateSecurityAssignedQueue(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.9.4 删除当前空间下分配的队列资源

### 功能介绍

删除当前空间下分配的队列资源。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/{project\_id}/security/permission/queue/assigned-source/{id}

表 12-238 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 需要删除的当前空间队列资源id。                         |

### 请求参数

表 12-239 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 400

表 12-240 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

无

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class DeleteSecurityAssignedQueueSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteSecurityAssignedQueueRequest request = new DeleteSecurityAssignedQueueRequest();
```

```
request.withId("{id}");
try {
    DeleteSecurityAssignedQueueResponse response = client.deleteSecurityAssignedQueue(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecurityAssignedQueueRequest()
        request.id = "{id}"
        response = client.delete_security_assigned_queue(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteSecurityAssignedQueueRequest{}
request.Id = "{id}"
response, err := client.DeleteSecurityAssignedQueue(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.10 数据分类接口

### 12.10.1 导入预置分类

#### 功能介绍

导入预置分类。

#### 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/{project\_id}/security/data-category/import-builtin-category

表 12-241 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-242 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                                |
|--------------|------|--------|-----------------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                                      |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。            |
| X-Language   | 是    | String | 请求语言 <ul style="list-style-type: none"> <li>zh-cn 中文</li> <li>en-us 英文</li> </ul> |

表 12-243 请求 Body 参数

| 参数                       | 是否必选 | 参数类型                                                       | 描述       |
|--------------------------|------|------------------------------------------------------------|----------|
| rule_secretcy_level_list | 否    | Array of <a href="#">ImportRuleSecrecyLevelDto</a> objects | 规则的密级列表。 |

表 12-244 ImportRuleSecrecyLevelDto

| 参数              | 是否必选 | 参数类型   | 描述                                    |
|-----------------|------|--------|---------------------------------------|
| builtin_rule_id | 否    | String | 内置规则模板id。                             |
| secretcy_level  | 否    | String | 密级id，获取方法请参见 <a href="#">获取数据密级</a> 。 |

## 响应参数

状态码： 200

表 12-245 响应 Body 参数

| 参数     | 参数类型                                                   | 描述    |
|--------|--------------------------------------------------------|-------|
| result | Array of <b>ImportDefaultCategoryResultDto</b> objects | 导入结果。 |

表 12-246 ImportDefaultCategoryResultDto

| 参数                   | 参数类型                                                   | 描述                                                                                           |
|----------------------|--------------------------------------------------------|----------------------------------------------------------------------------------------------|
| import_status        | String                                                 | 导入状态 <ul style="list-style-type: none"> <li>• success 导入成功</li> <li>• failed 导入失败</li> </ul> |
| import_error_message | String                                                 | 导入错误原因。                                                                                      |
| children             | Array of <b>ImportDefaultCategoryResultDto</b> objects | 子分类导入结果。                                                                                     |
| rule_result          | Array of <b>ImportDefaultRuleResultDto</b> objects     | 此分类绑定的规则导入的结果。                                                                               |
| uuid                 | String                                                 | 数据分类id。                                                                                      |
| name                 | String                                                 | 数据分类名称。                                                                                      |
| description          | String                                                 | 数据分类描述。                                                                                      |

表 12-247 ImportDefaultRuleResultDto

| 参数            | 参数类型   | 描述                                                                                           |
|---------------|--------|----------------------------------------------------------------------------------------------|
| import_status | String | 导入状态 <ul style="list-style-type: none"> <li>• success 导入成功</li> <li>• failed 导入失败</li> </ul> |



| 参数                              | 参数类型   | 描述                                                                                              |
|---------------------------------|--------|-------------------------------------------------------------------------------------------------|
| import_error_message            | String | 导入错误原因。                                                                                         |
| uuid                            | String | 内置规则模板id。                                                                                       |
| import_data_classification_rule | Object | 导入成功的识别规则。                                                                                      |
| rule_name                       | String | 数据识别规则名称。                                                                                       |
| rule_type                       | String | 数据识别规则类型 <ul style="list-style-type: none"> <li>• REGEX 正则表达式</li> <li>• KEYWORD 关键字</li> </ul> |
| rule_desc                       | String | 规则描述。                                                                                           |
| rule_name_en                    | String | 英文名称。                                                                                           |
| rule_desc_en                    | String | 英文描述。                                                                                           |
| country                         | String | 规则所属地区。                                                                                         |

表 12-248 ImportDataClassificationRuleDto

| 参数                  | 参数类型    | 描述                                                                                                               |
|---------------------|---------|------------------------------------------------------------------------------------------------------------------|
| uuid                | String  | 数据识别规则id。                                                                                                        |
| classification_type | String  | 识别规则类型 <ul style="list-style-type: none"> <li>• BUILTIN 内置</li> <li>• CUSTOM 自定义</li> </ul>                      |
| secrecy_level       | String  | 数据密级id。                                                                                                          |
| name                | String  | 数据识别规则名称。                                                                                                        |
| enable              | Boolean | 是否启用。                                                                                                            |
| method              | String  | 识别规则类型 <ul style="list-style-type: none"> <li>• NONE 无</li> <li>• REGULAR 正则表达式</li> <li>• DEFAULT 默认</li> </ul> |
| description         | String  | 描述。                                                                                                              |
| category_id         | String  | 数据分类id。                                                                                                          |
| builtin_rule_id     | String  | 预置规则id。                                                                                                          |
| updated_by          | String  | 更新人。                                                                                                             |

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| update_at  | Long   | 更新时间。 |
| created_by | String | 创建人。  |
| create_at  | Long   | 创建时间。 |

状态码： 400

表 12-249 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-category/import-builtin-category
{
  "rule_secrecy_level_list": [ {
    "builtin_rule_id": "a5adcd6881f011e9a870765a9db3b202",
    "secrecy_level": "8a9480087df0657e017df070a7620000"
  } ]
}
```

## 响应示例

状态码： 200

OK

```
{
  "result": [ {
    "name": "个人信息",
    "children": [ ],
    "description": null,
    "import_error_message": null,
    "import_status": "success",
    "uuid": "17fc7b0053c246518ade09feca480000",
    "rule_result": [ {
      "uuid": "a5adcb7a81f011e9a870765a9db3b202",
      "country": "China",
      "import_data_classification_rule": {
        "builtin_rule_id": "a5adcb7a81f011e9a870765a9db3b202",
        "category_id": "d4e8d3a67f984bc49907046e8fe37db0",
        "classification_type": "BUILTIN",
        "created_at": 1716773669838,
        "created_by": "user1",
        "description": "中国身份证号",
        "enable": true,
        "method": "DEFAULT",
        "name": "中国身份证号",
        "secrecy_level": "8a9481b68f7a96e7018fb7ae6814021b",
        "updated_at": 1716773669838,
        "updated_by": "user1",

```

```

"uuid": "8a9481b68f7a96e7018fb7af4bce0221"
},
"import_error_message": null,
"import_status": "success",
"rule_desc": "中国身份证号",
"rule_desc_en": "ID card number in (China)",
"rule_name": "中国身份证号",
"rule_name_en": "ID_Card_Number_(China)",
"rule_type": "REGEX"
}]
}]
}

```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.10.2 查询数据分类列表

### 功能介绍

查询数据分类列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/data-category

表 12-250 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-251 Query 参数

| 参数     | 是否必选 | 参数类型    | 描述                       |
|--------|------|---------|--------------------------|
| limit  | 否    | Integer | 每页显示的条目数量。               |
| offset | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。 |

## 请求参数

表 12-252 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-253 响应 Body 参数

| 参数              | 参数类型                                             | 描述       |
|-----------------|--------------------------------------------------|----------|
| total           | Integer                                          | 数据分类总的数量 |
| category_groups | Array of <a href="#">DataCategoryDTO</a> objects | 数据分类列表   |

表 12-254 DataCategoryDTO

| 参数             | 参数类型                                                            | 描述     |
|----------------|-----------------------------------------------------------------|--------|
| category_id    | String                                                          | 分类id   |
| category_name  | String                                                          | 分类名称   |
| description    | String                                                          | 分类描述   |
| category_level | Integer                                                         | 分类层级   |
| rule_list      | Array of <a href="#">DataClassificationRuleQueryDTO</a> objects | 识别规则   |
| root_id        | String                                                          | 分类树根节点 |
| parent_id      | String                                                          | 父分类节点  |
| category_path  | String                                                          | 分类树路径  |

| 参数          | 参数类型                                             | 描述          |
|-------------|--------------------------------------------------|-------------|
| create_by   | String                                           | 创建者         |
| create_time | Integer                                          | 创建时间        |
| update_by   | String                                           | 更新者         |
| update_time | Integer                                          | 更新时间        |
| project_id  | String                                           | 项目id        |
| instance_id | String                                           | 实例id        |
| synchronize | Boolean                                          | 同步（默认都同步资产） |
| children    | Array of <a href="#">DataCategoryDTO</a> objects | 子分类         |

表 12-255 DataClassificationRuleQueryDTO

| 参数                 | 参数类型    | 描述                           |
|--------------------|---------|------------------------------|
| uuid               | String  | 规则ID。                        |
| rule_type          | String  | 规则类型, CUSTOM, BUILTIN        |
| secrecy_level      | String  | 密级名称。                        |
| secrecy_level_num  | Long    | 密级层级。                        |
| name               | String  | 规则名称。                        |
| guid               | String  | guid。                        |
| enable             | Boolean | 规则是否开启。                      |
| method             | String  | 规则方式, REGULAR, NONE, DEFAULT |
| content_expression | String  | 内容表达式。                       |
| column_expression  | String  | 列表表达式。                       |
| commit_expression  | String  | 备注表达式。                       |
| project_id         | String  | 项目ID。                        |
| description        | String  | 规则描述。                        |
| created_by         | String  | 策略创建人。                       |
| created_at         | Long    | 策略创建时间。                      |

| 参数              | 参数类型   | 描述      |
|-----------------|--------|---------|
| updated_by      | String | 策略更新人。  |
| updated_at      | Long   | 策略更新时间。 |
| builtin_rule_id | String | 内置规则ID。 |
| category_id     | String | 分类ID。   |
| instance_id     | String | 实例ID。   |
| match_type      | String | 匹配类型。   |

状态码： 400

表 12-256 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/data-category
```

## 响应示例

状态码： 200

OK

```
{
  "category_groups": [ {
    "category_id": "555fb01bfcf04fed8623f6cfe81ef9b8",
    "category_level": 0,
    "category_name": "测试分类名",
    "category_path": "测试分类名",
    "children": [ ],
    "create_by": "chenxiaoyu",
    "create_time": 1717055180592,
    "description": "this is a example catagory",
    "instance_id": "dd97167b873d4a79b2aad54d4370a3bc",
    "parent_id": "-1,",
    "project_id": "0833a5737480d53b2f25c010dc1a7b88",
    "root_id": "555fb01bfcf04fed8623f6cfe81ef9b8",
    "rule_list": null,
    "synchronize": false,
    "update_by": null,
    "update_time": null
  } ],
  "total": 1
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.11 数据安全诊断接口

### 12.11.1 执行数据安全诊断

#### 功能介绍

执行数据安全诊断。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/{project\_id}/security/diagnose/rescan

表 12-257 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

#### 请求参数

表 12-258 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-259 请求 Body 参数

| 参数   | 是否必选 | 参数类型   | 描述                                                                                                                                                               |
|------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type | 否    | String | 数据安全诊断项 <ul style="list-style-type: none"> <li>• SENSITIVE_DATA 敏感数据保护</li> <li>• PERMISSION_MANAGEMENT 数据权限控制</li> <li>• DATASOURCE_PROTECTION 数据源防护</li> </ul> |

## 响应参数

状态码： 400

表 12-260 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/diagnose/rescan
{
  "type": "SENSITIVE_DATA"
}
```

## 响应示例

无

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.11.2 查询不合理的权限配置

### 功能介绍

查询不合理的权限配置。



## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/security/diagnose/unreasonable-permissions

表 12-261 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-262 Query 参数

| 参数              | 是否必选 | 参数类型    | 描述                                                                                                    |
|-----------------|------|---------|-------------------------------------------------------------------------------------------------------|
| limit           | 否    | Integer | 每页显示的条目数量。                                                                                            |
| offset          | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。                                                                              |
| diagnose_id     | 是    | String  | 诊断任务id，通过调用查询数据权限控制模块诊断结果接口获取。                                                                        |
| datasource_type | 否    | String  | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| user_name       | 否    | String  | 用户名。                                                                                                  |

## 请求参数

表 12-263 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-264 响应 Body 参数

| 参数                       | 参数类型                                             | 描述          |
|--------------------------|--------------------------------------------------|-------------|
| total                    | Integer                                          | 不合理权限配置总条数。 |
| unreasonable_permissions | Array of <b>DiagnosePermissionDetail</b> objects | 不合理权限配置列表。  |

表 12-265 DiagnosePermissionDetail

| 参数                | 参数类型   | 描述       |
|-------------------|--------|----------|
| id                | String | 权限配置编号。  |
| task_id           | String | 诊断任务id。  |
| user_name         | String | 用户名。     |
| permission_type   | String | 权限类型。    |
| permission_action | String | 权限操作。    |
| permission_source | String | 权限来源。    |
| datasource_type   | String | 数据源类型。   |
| cluster_name      | String | 集群名称。    |
| database          | String | 数据库名。    |
| schema            | String | schema名。 |
| table             | String | 表名。      |
| remark            | String | 备注。      |

状态码： 400

表 12-266 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/diagnose/unreasonable-permissions?limit=10&offset=0&diagnose_id=6f28400297cc4402adfb13ce7533f996
```

## 响应示例

状态码： 200

OK

```
{
  "total": 1,
  "unreasonable_permissions": [ {
    "cluster_name": "mrs_3x_autotest_do_not_del",
    "database": "default",
    "datasource_type": "HIVE",
    "id": "0d524dab271b4050be7da83eb9d0af85",
    "permission_action": "DROP",
    "permission_source": "拆分后的二层权限",
    "permission_type": "ALLOW",
    "remark": "Other roles except admin are not advised to have the DROP permission.",
    "schema": null,
    "table": "allfuhe",
    "task_id": "6f28400297cc4402adfb13ce7533f996",
    "user_name": "user1"
  } ]
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.11.3 查询数据权限控制模块诊断结果

### 功能介绍

查询数据权限控制模块诊断结果。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/security/diagnose/permission-management

表 12-267 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-268 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-269 响应 Body 参数

| 参数                      | 参数类型                                          | 描述           |
|-------------------------|-----------------------------------------------|--------------|
| task_id                 | String                                        | 诊断任务id。      |
| check_time              | Long                                          | 最新检测时间。      |
| scanning                | Boolean                                       | 是否正在诊断。      |
| high_permission         | <a href="#">HighPermission</a> object         | 高权限管理诊断结果。   |
| unreasonable_permission | <a href="#">UnreasonablePermission</a> object | 权限控制合理性检测结果。 |

表 12-270 HighPermission

| 参数                     | 参数类型   | 描述                                                                                                                                                  |
|------------------------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| result                 | String | 检测结果 <ul style="list-style-type: none"> <li>• NO_RISK 无风险</li> <li>• MEDIUM_RISK 中风险</li> <li>• HIGH_RISK 高风险</li> <li>• NOT_SCANNED 未扫描</li> </ul> |
| workspace_admin        | String | 空间管理员用户列表。                                                                                                                                          |
| security_administrator | String | 安全管理员用户列表。                                                                                                                                          |

表 12-271 UnreasonablePermission

| 参数     | 参数类型    | 描述                                                                                                                                                  |
|--------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| result | String  | 检测结果 <ul style="list-style-type: none"> <li>• NO_RISK 无风险</li> <li>• MEDIUM_RISK 中风险</li> <li>• HIGH_RISK 高风险</li> <li>• NOT_SCANNED 未扫描</li> </ul> |
| count  | Integer | 存在风险的权限控制数量。                                                                                                                                        |

状态码： 400

表 12-272 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

/v1/0833a5737480d53b2f25c010dc1a7b88/security/diagnose/permission-management

## 响应示例

状态码： 200

OK

```
{
  "check_time": 1703488426000,
  "high_permission": {
    "result": "MEDIUM_RISK",
    "security_administrator": "user1,user2,user3",
    "workspace_admin": "user1,user2,user3"
  },
  "scanning": false,
  "task_id": "6f28400297cc4402adfb13ce7533f996",
  "unreasonable_permission": {
    "count": 7,
    "result": "MEDIUM_RISK"
  }
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.11.4 查询敏感数据保护模块诊断结果

### 功能介绍

查询敏感数据保护模块诊断结果

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/diagnose/sensitive-data

表 12-273 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-274 请求 Header 参数

| 参数        | 是否必选 | 参数类型   | 描述                                           |
|-----------|------|--------|----------------------------------------------|
| workspace | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。 |

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-275 响应 Body 参数

| 参数             | 参数类型                                    | 描述         |
|----------------|-----------------------------------------|------------|
| task_id        | String                                  | 诊断任务id     |
| scanning       | Boolean                                 | 是否正在扫描     |
| check_time     | Long                                    | 最新检测时间。    |
| classification | <b>Classification Result</b> object     | 数据分级分类诊断结果 |
| rule           | <b>Identification RuleResult</b> object | 敏感识别规则诊断结果 |
| masking        | <b>DataMasking Result</b> object        | 数据脱敏诊断结果   |

表 12-276 ClassificationResult

| 参数             | 参数类型    | 描述                                                                                                                                                  |
|----------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| result         | String  | 检测结果 <ul style="list-style-type: none"> <li>• NO_RISK 无风险</li> <li>• MEDIUM_RISK 中风险</li> <li>• HIGH_RISK 高风险</li> <li>• NOT_SCANNED 未扫描</li> </ul> |
| security_level | Boolean | 是否配置了密级                                                                                                                                             |
| classification | Boolean | 是否配置了分类                                                                                                                                             |

表 12-277 IdentificationRuleResult

| 参数            | 参数类型                               | 描述                                                                                                                                                  |
|---------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| result        | String                             | 检测结果 <ul style="list-style-type: none"> <li>• NO_RISK 无风险</li> <li>• MEDIUM_RISK 中风险</li> <li>• HIGH_RISK 高风险</li> <li>• NOT_SCANNED 未扫描</li> </ul> |
| count         | Integer                            | 有风险的问题数量                                                                                                                                            |
| abnormal_info | Array of SecurityLevelInfo objects | 没有配置识别规则的密级列表                                                                                                                                       |

表 12-278 SecurityLevelInfo

| 参数    | 参数类型   | 描述   |
|-------|--------|------|
| name  | String | 密级名称 |
| level | String | 密级等级 |

表 12-279 DataMaskingResult

| 参数     | 参数类型    | 描述                                                                                                                                                  |
|--------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| result | String  | 检测结果 <ul style="list-style-type: none"> <li>• NO_RISK 无风险</li> <li>• MEDIUM_RISK 中风险</li> <li>• HIGH_RISK 高风险</li> <li>• NOT_SCANNED 未扫描</li> </ul> |
| count  | Integer | 没有配置脱敏任务的表数量                                                                                                                                        |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/diagnose/sensitive-data
```

## 响应示例

状态码: 200

OK

```
{
  "check_time": 1716435728000,
  "classification": {
```



```

"classification" : true,
"result" : "NO_RISK",
"security_level" : true
},
"masking" : {
"count" : 0,
"result" : "NO_RISK"
},
"rule" : {
"abnormal_info" : [ {
"level" : "3",
"name" : "yyy"
}, {
"level" : "8",
"name" : "高度绝密"
} ],
"count" : 2,
"result" : "HIGH_RISK"
},
"scanning" : false,
"task_id" : "695e9ea4dd864e55b5dbdd91ad61ebd0"
}
    
```

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 12.11.5 查询未进行静态脱敏任务的表信息

### 功能介绍

查询未进行静态脱敏任务的表信息

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/diagnose/masking-result

表 12-280 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-281 Query 参数

| 参数              | 是否必选 | 参数类型    | 描述                                                                                                    |
|-----------------|------|---------|-------------------------------------------------------------------------------------------------------|
| limit           | 否    | Integer | 每页显示的条目数量。                                                                                            |
| offset          | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。                                                                              |
| diagnose_id     | 是    | String  | 诊断任务id，通过调用查询数据权限控制模块诊断结果接口获取。                                                                        |
| datasource_type | 否    | String  | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| database        | 否    | String  | 数据库名称                                                                                                 |
| cluster_name    | 否    | String  | 集群名称。                                                                                                 |

## 请求参数

表 12-282 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-283 响应 Body 参数

| 参数     | 参数类型                                                     | 描述     |
|--------|----------------------------------------------------------|--------|
| total  | Integer                                                  | 表总数    |
| tables | Array of <a href="#">DiagnoseNoMaskingDetail</a> objects | 查询的表集合 |

表 12-284 DiagnoseNoMaskingDetail

| 参数              | 参数类型   | 描述       |
|-----------------|--------|----------|
| id              | String | 详情uuid   |
| project_id      | String | 项目ID     |
| task_id         | String | 诊断任务id   |
| datasource_type | String | 数据源类型    |
| cluster_name    | String | 集群名称     |
| database        | String | 数据库名称    |
| schema          | String | schema名称 |
| table           | String | 表名称      |
| remark          | String | 详情评价     |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/diagnose/masking-result?
limit=10&offset=0&diagnose_id=695e9ea4dd864e55b5dbdd91ad61ebd0
```

## 响应示例

状态码： 200

OK

```
{
  "total": 1,
  "tables": [ {
    "id": "02520cbff97c58ed963844bf5590bde",
    "project_id": "0833a5737480d53b2f25c010dc1a7b88",
    "task_id": "eb0e436378924a05878029e0b153a257",
    "datasource_type": "DWS",
    "cluster_name": "dws",
    "database": "test1",
    "schema": "public",
    "table": "table",
    "remark": "remark"
  } ]
}
```

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 12.11.6 查询数据源防护模块诊断结果

### 功能介绍

查询数据源防护模块诊断结果

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/diagnose/datasource-protection

表 12-285 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-286 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

### 响应参数

状态码： 200

表 12-287 响应 Body 参数

| 参数         | 参数类型    | 描述      |
|------------|---------|---------|
| task_id    | String  | 诊断任务id  |
| scanning   | Boolean | 是否正在扫描  |
| check_time | Long    | 最新检测时间。 |

| 参数                    | 参数类型                                | 描述       |
|-----------------------|-------------------------------------|----------|
| kerberos              | <b>SecurityCertification</b> object | 安全认证诊断结果 |
| public_network_access | <b>PublicNetworkAccess</b> object   | 公网访问诊断结果 |
| security_group        | <b>SecurityGroupResult</b> object   | 安全组诊断结果  |

表 12-288 SecurityCertification

| 参数            | 参数类型                                   | 描述                                                                                                                                                  |
|---------------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| result        | String                                 | 检测结果 <ul style="list-style-type: none"> <li>• NO_RISK 无风险</li> <li>• MEDIUM_RISK 中风险</li> <li>• HIGH_RISK 高风险</li> <li>• NOT_SCANNED 未扫描</li> </ul> |
| kerberos_info | Array of <b>KerberosStatus</b> objects | kerberos信息                                                                                                                                          |

表 12-289 KerberosStatus

| 参数           | 参数类型    | 描述                                 |
|--------------|---------|------------------------------------|
| cluster_id   | String  | 集群id                               |
| cluster_name | String  | 集群名称                               |
| status       | Boolean | 是否开启了kerberos认证,true:开启, false:未开启 |

表 12-290 PublicNetworkAccess

| 参数     | 参数类型   | 描述                                                                                                                                                  |
|--------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| result | String | 检测结果 <ul style="list-style-type: none"> <li>• NO_RISK 无风险</li> <li>• MEDIUM_RISK 中风险</li> <li>• HIGH_RISK 高风险</li> <li>• NOT_SCANNED 未扫描</li> </ul> |

| 参数                  | 参数类型                                        | 描述         |
|---------------------|---------------------------------------------|------------|
| public_network_info | Array of <b>PublicNetworkStatus</b> objects | kerberos信息 |

表 12-291 PublicNetworkStatus

| 参数           | 参数类型    | 描述                           |
|--------------|---------|------------------------------|
| cluster_id   | String  | 集群id                         |
| cluster_name | String  | 集群名称                         |
| status       | Boolean | 是否开启了公网访问,true:开启, false:未开启 |
| ip           | String  | 公网ip地址                       |

表 12-292 SecurityGroupResult

| 参数             | 参数类型                                        | 描述                                                                                                                                                  |
|----------------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| result         | String                                      | 检测结果 <ul style="list-style-type: none"> <li>• NO_RISK 无风险</li> <li>• MEDIUM_RISK 中风险</li> <li>• HIGH_RISK 高风险</li> <li>• NOT_SCANNED 未扫描</li> </ul> |
| security_group | Array of <b>SecurityGroupStatus</b> objects | kerberos信息                                                                                                                                          |

表 12-293 SecurityGroupStatus

| 参数                  | 参数类型   | 描述    |
|---------------------|--------|-------|
| cluster_id          | String | 集群id  |
| cluster_name        | String | 集群名称  |
| security_group_name | String | 安全组名称 |
| group_description   | String | 风险说明  |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/diagnose/datasource-protection
```

## 响应示例

状态码： 200

OK

```
{
  "check_time": 1716455965000,
  "kerberos": {
    "kerberos_info": [ {
      "cluster_id": "ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
      "cluster_name": "mrs_noauth_autotest_do_not_del",
      "status": false
    } ],
    "result": "MEDIUM_RISK"
  },
  "public_network_access": {
    "public_network_info": [ {
      "cluster_id": "ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
      "cluster_name": "mrs_noauth_autotest_do_not_del",
      "ip": "100.93.6.203",
      "status": true
    } ],
    "result": "MEDIUM_RISK"
  },
  "scanning": false,
  "security_group": {
    "result": "MEDIUM_RISK",
    "security_group": [ {
      "cluster_id": "ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
      "cluster_name": "mrs_noauth_autotest_do_not_del",
      "group_description": "default",
      "security_group_name": "default"
    } ]
  },
  "task_id": "bd10160985ff47dd8f1edf8115a3cec3"
}
```

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 12.12 空间资源权限策略管理接口

### 12.12.1 查询空间资源权限策略列表

#### 功能介绍

查询空间资源权限策略列表

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/{project\_id}/security/permission-resource

表 12-294 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-295 Query 参数

| 参数            | 是否必选 | 参数类型    | 描述                       |
|---------------|------|---------|--------------------------|
| limit         | 否    | Integer | 每页显示的条目数量。               |
| offset        | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。 |
| policy_name   | 否    | String  | 策略名称。                    |
| resource_name | 否    | String  | 授权资源名称。                  |
| member_name   | 否    | String  | 成员名称。                    |
| order_by      | 否    | String  | 排序参数，NAME,UPDATE_TIME。   |
| order_by_asc  | 否    | Boolean | 是否升序（仅指定排序参数时有效）。        |

## 请求参数

表 12-296 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200



表 12-297 响应 Body 参数

| 参数       | 参数类型                                             | 描述          |
|----------|--------------------------------------------------|-------------|
| total    | Integer                                          | 空间资源权限策略总条数 |
| policies | Array of <b>PermissionResourcePolicy</b> objects | 空间资源权限策略列表  |

表 12-298 PermissionResourcePolicy

| 参数          | 参数类型                                       | 描述     |
|-------------|--------------------------------------------|--------|
| policy_id   | String                                     | 策略id   |
| policy_name | String                                     | 策略名称   |
| resources   | Array of <b>ResourcePolicyItem</b> objects | 资源对象列表 |
| members     | Array of <b>MemberPolicyItem</b> objects   | 成员列表   |
| create_time | Long                                       | 创建时间   |
| create_user | String                                     | 创建用户   |
| update_time | Long                                       | 修改时间   |

表 12-299 ResourcePolicyItem

| 参数            | 参数类型   | 描述                          |
|---------------|--------|-----------------------------|
| resource_id   | String | 资源id                        |
| resource_name | String | 资源名称                        |
| resource_type | String | 资源类型:DATA_CONNECTION,AGENCY |

表 12-300 MemberPolicyItem

| 参数        | 参数类型   | 描述   |
|-----------|--------|------|
| member_id | String | 成员id |

| 参数          | 参数类型   | 描述                                                       |
|-------------|--------|----------------------------------------------------------|
| member_name | String | 成员名称                                                     |
| member_type | String | 成员类型:USER,USER_GROUP,WORKSPACE_ROLE, 分别代表空间用户、空间用户组、空间角色 |

状态码: 400

表 12-301 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-resource?offset=0&limit=100&policy_name=TEST_POLICY_NAME
```

```
{
  "limit": 20,
  "offset": 0
}
```

## 响应示例

状态码: 200

OK

```
{
  "policies": [ {
    "create_time": 1698202688000,
    "create_user": "ei_dayu_test_01",
    "policy_id": "fea96c90024711b8bf8d6886407b814b",
    "policy_name": "test_permission_resource",
    "update_time": 1698202688000,
    "members": [ {
      "member_id": "fea96c90024711b8bf8d6886407b814b",
      "member_name": "common_user",
      "member_type": "USER"
    }, {
      "member_id": "0833a5736980d53b0f22c0102ffc0102",
      "member_name": "ei_dayu_test_usergroup_01",
      "member_type": "USER_GROUP"
    } ],
    "resources": [ {
      "resource_id": "7c8a2d85d917492bb3195377cd9c36be",
      "resource_name": "hive",
      "resource_type": "DATA_CONNECTION"
    } ]
  } ],
  "total": 1
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.12.2 创建空间资源权限策略

### 功能介绍

创建空间资源权限策略

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-resource

表 12-302 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-303 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-304 请求 Body 参数

| 参数          | 是否必选 | 参数类型                                | 描述                                                                                                           |
|-------------|------|-------------------------------------|--------------------------------------------------------------------------------------------------------------|
| policy_name | 是    | String                              | 策略名称：英文和汉字开头，支持英文、汉字、数字、下划线，2-64字符                                                                           |
| resources   | 是    | Array of ResourcePolicyItem objects | 资源对象列表。资源对象包含数据连接，连接获取方法详见 <a href="#">查询数据连接列表</a>                                                          |
| members     | 是    | Array of MemberPolicyItem objects   | 成员列表。成员包含空间用户、空间用户组、空间用户角色。空间用户、用户组获取方法请参见 <a href="#">获取工作空间用户信息</a> ，空间角色获取方法参见 <a href="#">获取工作空间用户角色</a> |

表 12-305 ResourcePolicyItem

| 参数            | 是否必选 | 参数类型   | 描述                          |
|---------------|------|--------|-----------------------------|
| resource_id   | 是    | String | 资源id                        |
| resource_name | 是    | String | 资源名称                        |
| resource_type | 是    | String | 资源类型:DATA_CONNECTION,AGENCY |

表 12-306 MemberPolicyItem

| 参数          | 是否必选 | 参数类型   | 描述                                                      |
|-------------|------|--------|---------------------------------------------------------|
| member_id   | 是    | String | 成员id                                                    |
| member_name | 是    | String | 成员名称                                                    |
| member_type | 否    | String | 成员类型:USER,USER_GROUP,WORKSPACE_ROLE，分别代表空间用户、空间用户组、空间角色 |

## 响应参数

状态码： 200

表 12-307 响应 Body 参数

| 参数          | 参数类型                                | 描述     |
|-------------|-------------------------------------|--------|
| policy_id   | String                              | 策略id   |
| policy_name | String                              | 策略名称   |
| resources   | Array of ResourcePolicyItem objects | 资源对象列表 |
| members     | Array of MemberPolicyItem objects   | 成员列表   |
| create_time | Long                                | 创建时间   |
| create_user | String                              | 创建用户   |
| update_time | Long                                | 修改时间   |

表 12-308 ResourcePolicyItem

| 参数            | 参数类型   | 描述                          |
|---------------|--------|-----------------------------|
| resource_id   | String | 资源id                        |
| resource_name | String | 资源名称                        |
| resource_type | String | 资源类型:DATA_CONNECTION,AGENCY |

表 12-309 MemberPolicyItem

| 参数          | 参数类型   | 描述                                                       |
|-------------|--------|----------------------------------------------------------|
| member_id   | String | 成员id                                                     |
| member_name | String | 成员名称                                                     |
| member_type | String | 成员类型:USER,USER_GROUP,WORKSPACE_ROLE, 分别代表空间用户、空间用户组、空间角色 |

状态码: 400

表 12-310 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-resource
{
  "policy_name": "test_resource_permission_policy",
  "members": [ {
    "member_id": "fea96c90024711b8bf8d6886407b814b",
    "member_name": "common_user",
    "member_type": "USER"
  }, {
    "member_id": "0833a5736980d53b0f22c0102ffc0",
    "member_name": "ei_dayu_test_usergroup_01",
    "member_type": "USER_GROUP"
  } ],
  "resources": [ {
    "resource_id": "7c8a2d85d917492bb3195377cd9c36be",
    "resource_name": "hive",
    "resource_type": "DATA_CONNECTION"
  } ]
}
```

## 响应示例

状态码: 200

OK

```
{
  "policy_id": "b43927a45a514e93ba70524b28923500",
  "policy_name": "test_dataconnection_permission_policy",
  "create_time": 1698202688247,
  "update_time": 1698202688247,
  "create_user": "ei_dayu_test_01",
  "members": [ {
    "member_id": "fea96c90024711b8bf8d6886407b814b",
    "member_name": "common_user",
    "member_type": "USER"
  }, {
    "member_id": "0833a5736980d53b0f22c0102ffc0",
    "member_name": "ei_dayu_test_usergroup_01",
    "member_type": "USER_GROUP"
  } ],
  "resources": [ {
    "resource_id": "7c8a2d85d917492bb3195377cd9c36be",
    "resource_name": "hive",
    "resource_type": "DATA_CONNECTION"
  } ]
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.12.3 查询单个资源权限策略

### 功能介绍

查询单个资源权限策略

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/permission-resource/{policy\_id}

表 12-311 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| policy_id  | 是    | String | 空间资源权限策略id。                              |

### 请求参数

表 12-312 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

### 响应参数

状态码： 200

表 12-313 响应 Body 参数

| 参数          | 参数类型                                | 描述     |
|-------------|-------------------------------------|--------|
| policy_id   | String                              | 策略id   |
| policy_name | String                              | 策略名称   |
| resources   | Array of ResourcePolicyItem objects | 资源对象列表 |
| members     | Array of MemberPolicyItem objects   | 成员列表   |
| create_time | Long                                | 创建时间   |
| create_user | String                              | 创建用户   |
| update_time | Long                                | 修改时间   |

表 12-314 ResourcePolicyItem

| 参数            | 参数类型   | 描述                          |
|---------------|--------|-----------------------------|
| resource_id   | String | 资源id                        |
| resource_name | String | 资源名称                        |
| resource_type | String | 资源类型:DATA_CONNECTION,AGENCY |

表 12-315 MemberPolicyItem

| 参数          | 参数类型   | 描述                                                       |
|-------------|--------|----------------------------------------------------------|
| member_id   | String | 成员id                                                     |
| member_name | String | 成员名称                                                     |
| member_type | String | 成员类型:USER,USER_GROUP,WORKSPACE_ROLE, 分别代表空间用户、空间用户组、空间角色 |

状态码: 400



表 12-316 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-resource/578f83559cdb4cebb0ff4527b59a2fda
```

## 响应示例

状态码： 200

OK

```
{
  "policy_id": "b43927a45a514e93ba70524b28923500",
  "policy_name": "test_dataconnection_permission_policy",
  "create_time": 1698202688247,
  "update_time": 1698202688247,
  "create_user": "ei_dayu_test_01",
  "members": [ {
    "member_id": "fea96c90024711b8bf8d6886407b814b",
    "member_name": "common_user",
    "member_type": "USER"
  }, {
    "member_id": "0833a5736980d53b0f22c0102ffc0",
    "member_name": "ei_dayu_test_usergroup_01",
    "member_type": "USER_GROUP"
  } ],
  "resources": [ {
    "resource_id": "7c8a2d85d917492bb3195377cd9c36be",
    "resource_name": "hive",
    "resource_type": "DATA_CONNECTION"
  } ]
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.12.4 更新空间资源权限策略

### 功能介绍

更新空间资源权限策略

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1/{project\_id}/security/permission-resource/{policy\_id}

表 12-317 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| policy_id  | 是    | String | 空间资源权限策略id。                              |

## 请求参数

表 12-318 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-319 请求 Body 参数

| 参数          | 是否必选 | 参数类型                                                | 描述                                                                                                           |
|-------------|------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| policy_name | 是    | String                                              | 策略名称：英文和汉字开头，支持英文、汉字、数字、下划线，2-64字符                                                                           |
| resources   | 是    | Array of <a href="#">ResourcePolicyItem</a> objects | 资源对象列表。资源对象包含数据连接，连接获取方法详见 <a href="#">查询数据连接列表</a>                                                          |
| members     | 是    | Array of <a href="#">MemberPolicyItem</a> objects   | 成员列表。成员包含空间用户、空间用户组、空间用户角色。空间用户、用户组获取方法请参见 <a href="#">获取工作空间用户信息</a> ，空间角色获取方法参见 <a href="#">获取工作空间用户角色</a> |

表 12-320 ResourcePolicyItem

| 参数            | 是否必选 | 参数类型   | 描述                          |
|---------------|------|--------|-----------------------------|
| resource_id   | 是    | String | 资源id                        |
| resource_name | 是    | String | 资源名称                        |
| resource_type | 是    | String | 资源类型:DATA_CONNECTION,AGENCY |

表 12-321 MemberPolicyItem

| 参数          | 是否必选 | 参数类型   | 描述                                                       |
|-------------|------|--------|----------------------------------------------------------|
| member_id   | 是    | String | 成员id                                                     |
| member_name | 是    | String | 成员名称                                                     |
| member_type | 否    | String | 成员类型:USER,USER_GROUP,WORKSPACE_ROLE, 分别代表空间用户、空间用户组、空间角色 |

## 响应参数

状态码： 200

表 12-322 响应 Body 参数

| 参数          | 参数类型                                                | 描述     |
|-------------|-----------------------------------------------------|--------|
| policy_id   | String                                              | 策略id   |
| policy_name | String                                              | 策略名称   |
| resources   | Array of <a href="#">ResourcePolicyItem</a> objects | 资源对象列表 |
| members     | Array of <a href="#">MemberPolicyItem</a> objects   | 成员列表   |
| create_time | Long                                                | 创建时间   |
| create_user | String                                              | 创建用户   |
| update_time | Long                                                | 修改时间   |

表 12-323 ResourcePolicyItem

| 参数            | 参数类型   | 描述                          |
|---------------|--------|-----------------------------|
| resource_id   | String | 资源id                        |
| resource_name | String | 资源名称                        |
| resource_type | String | 资源类型:DATA_CONNECTION,AGENCY |

表 12-324 MemberPolicyItem

| 参数          | 参数类型   | 描述                                                       |
|-------------|--------|----------------------------------------------------------|
| member_id   | String | 成员id                                                     |
| member_name | String | 成员名称                                                     |
| member_type | String | 成员类型:USER,USER_GROUP,WORKSPACE_ROLE, 分别代表空间用户、空间用户组、空间角色 |

状态码： 400

表 12-325 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-resource/578f83559cdb4cebb0ff4527b59a2fda
```

```
{
  "policy_name": "test_resource_permission_policy",
  "members": [ {
    "member_id": "fea96c90024711b8bf8d6886407b814b",
    "member_name": "common_user",
    "member_type": "USER"
  }, {
    "member_id": "0833a5736980d53b0f22c0102ffc0",
    "member_name": "ei_dayu_test_usergroup_01",
    "member_type": "USER_GROUP"
  } ],
  "resources": [ {
    "resource_id": "7c8a2d85d917492bb3195377cd9c36be",
```

```
"resource_name": "hive",  
"resource_type": "DATA_CONNECTION"  
}]  
}
```

## 响应示例

状态码: 200

OK

```
{  
  "policy_id": "b43927a45a514e93ba70524b28923500",  
  "policy_name": "test_dataconnection_permission_policy",  
  "create_time": 1698202688247,  
  "update_time": 1698202688247,  
  "create_user": "ei_dayu_test_01",  
  "members": [{  
    "member_id": "fea96c90024711b8bf8d6886407b814b",  
    "member_name": "common_user",  
    "member_type": "USER"  
  }, {  
    "member_id": "0833a5736980d53b0f22c0102ffcbfc0",  
    "member_name": "ei_dayu_test_usergroup_01",  
    "member_type": "USER_GROUP"  
  }],  
  "resources": [{  
    "resource_id": "7c8a2d85d917492bb3195377cd9c36be",  
    "resource_name": "hive",  
    "resource_type": "DATA_CONNECTION"  
  }]  
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.12.5 批量删除资源权限策略

### 功能介绍

批量删除资源权限策略

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-resource/batch-delete

表 12-326 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-327 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-328 请求 Body 参数

| 参数    | 是否必选 | 参数类型             | 描述      |
|-------|------|------------------|---------|
| dw_id | 否    | String           | 数据连接id。 |
| ids   | 否    | Array of strings | id列表。   |

## 响应参数

状态码： 400

表 12-329 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-resource/578f83559cdb4cebb0ff4527b59a2fda
{
```

```
"ids" : [ "d9ff4b06db43e3d4f81de1e60077a480" ]  
}
```

## 响应示例

无

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

# 12.13 安全管理员接口

## 12.13.1 查看安全管理员

### 功能介绍

查看安全管理员。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/admin

表 12-330 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-331 请求 Header 参数

| 参数        | 是否必选 | 参数类型   | 描述                                           |
|-----------|------|--------|----------------------------------------------|
| workspace | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。 |

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-332 响应 Body 参数

| 参数           | 参数类型   | 描述                                                    |
|--------------|--------|-------------------------------------------------------|
| manager_type | String | 管理员类型, USER:用户, USER_GROUP:用户组                        |
| manager_id   | String | 管理员id, 管理员类型为用户时, 请传入iam用户id; 管理员类型为用户组时, 请传入iam用户组id |
| manager_name | String | 管理员名称, 管理员类型为用户时, 请传入iam用户名称; 管理员类型为用户组时, 请传入iam用户组名称 |

状态码： 400

表 12-333 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/admin
```

## 响应示例

状态码： 200

OK

```
{
  "manager_type": "USER",
  "manager_id": "0833a5737480d53b2f25c010dc1a7b88",
  "manager_name": "test_user"
}
```



## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.13.2 创建或更新安全管理员

### 功能介绍

创建或更新安全管理员。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/admin

表 12-334 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-335 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-336 请求 Body 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                    |
|--------------|------|--------|-------------------------------------------------------|
| manager_type | 否    | String | 管理员类型, USER:用户, USER_GROUP:用户组                        |
| manager_id   | 否    | String | 管理员id, 管理员类型为用户时, 请传入iam用户id; 管理员类型为用户组时, 请传入iam用户组id |
| manager_name | 否    | String | 管理员名称, 管理员类型为用户时, 请传入iam用户名称; 管理员类型为用户组时, 请传入iam用户组名称 |

## 响应参数

状态码: 200

表 12-337 响应 Body 参数

| 参数           | 参数类型   | 描述                                                    |
|--------------|--------|-------------------------------------------------------|
| manager_type | String | 管理员类型, USER:用户, USER_GROUP:用户组                        |
| manager_id   | String | 管理员id, 管理员类型为用户时, 请传入iam用户id; 管理员类型为用户组时, 请传入iam用户组id |
| manager_name | String | 管理员名称, 管理员类型为用户时, 请传入iam用户名称; 管理员类型为用户组时, 请传入iam用户组名称 |

状态码: 400

表 12-338 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/admin
{
  "manager_type": "USER",
  "manager_id": "0833a5737480d53b2f25c010dc1a7b88",
  "manager_name": "test_user"
}
```

## 响应示例

状态码： 200

OK

```
{  
  "manager_type": "USER",  
  "manager_id": "0833a5737480d53b2f25c010dc1a7b88",  
  "manager_name": "test_user"  
}
```

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.14 动态数据脱敏接口

### 12.14.1 查询动态数据脱敏策略列表

#### 功能介绍

查询动态数据脱敏策略列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/{project\_id}/security/masking/dynamic/policies

表 12-339 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-340 Query 参数

| 参数    | 是否必选 | 参数类型    | 描述         |
|-------|------|---------|------------|
| limit | 否    | Integer | 每页显示的条目数量。 |

| 参数            | 是否必选 | 参数类型    | 描述                       |
|---------------|------|---------|--------------------------|
| offset        | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。 |
| name          | 否    | String  | 动态脱敏策略名称。                |
| cluster_name  | 否    | String  | 集群名称。                    |
| database_name | 否    | String  | 数据库名称。                   |
| table_name    | 否    | String  | 表名称。                     |
| order_by      | 否    | String  | 排序参数，UPDATE_TIME。        |
| order_by_asc  | 否    | Boolean | 是否升序（仅指定排序参数时有效）。        |

## 请求参数

表 12-341 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-342 响应 Body 参数

| 参数       | 参数类型                                                     | 描述          |
|----------|----------------------------------------------------------|-------------|
| total    | Integer                                                  | 动态脱敏策略总条数。  |
| policies | Array of <a href="#">DynamicMaskingPolicySet</a> objects | 动态数据脱敏策略列表。 |

表 12-343 DynamicMaskingPolicySet

| 参数              | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                        |
|-----------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id              | String | 动态脱敏策略id。                                                                                                                                                                                                                                                                                                                 |
| name            | String | 动态脱敏策略名称。英文和汉字开头, 支持英文、汉字、数字、下划线, 2-64字符。                                                                                                                                                                                                                                                                                 |
| datasource_type | String | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul>                                                                                                                                                                                                                     |
| cluster_id      | String | 集群id。请于集群管理页面查看集群ID信息。当数据源类型为DLI时, 该参数需要填写为DLI。                                                                                                                                                                                                                                                                           |
| cluster_name    | String | 集群名称。请于集群管理页面查看集群名称信息。当数据源类型为DLI时, 该参数需要填写为DLI。                                                                                                                                                                                                                                                                           |
| database_name   | String | 数据库名称。获取方法请参见获取数据源中的表。                                                                                                                                                                                                                                                                                                    |
| table_name      | String | 数据表名称, 获取方法请参见获取数据源中的表。                                                                                                                                                                                                                                                                                                   |
| user_groups     | String | 用户组列表, 用户组名称逗号分隔 (非必填项, 但用户、用户组必须二选其一进行配置)。例如: "userGroup1,userGroup2"。                                                                                                                                                                                                                                                   |
| users           | String | 用户列表, 用户名称逗号分隔 (非必填项, 但用户、用户组必须二选其一进行配置), 例如: "user1,user2"。                                                                                                                                                                                                                                                              |
| sync_status     | String | 同步状态: <ul style="list-style-type: none"> <li>• UNKNOWN 未知状态</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> <li>• SYNC_PARTIAL_FAIL 存在失败</li> <li>• DELETE_FAIL 删除失败</li> <li>• DELETING 删除中</li> <li>• UPDATING 更新中</li> <li>• DATA_UPDATED 数据存在更新</li> </ul> |
| sync_time       | Long   | 策略同步时间。                                                                                                                                                                                                                                                                                                                   |
| sync_msg        | String | 策略同步日志。                                                                                                                                                                                                                                                                                                                   |
| create_time     | Long   | 策略创建时间。                                                                                                                                                                                                                                                                                                                   |

| 参数          | 参数类型   | 描述      |
|-------------|--------|---------|
| create_user | String | 策略创建者。  |
| update_time | Long   | 策略更新时间。 |
| update_user | String | 策略更新者。  |

状态码： 400

表 12-344 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

无

## 响应示例

状态码： 200

OK

```
{
  "policies": [ {
    "cluster_id": "ec26b187-626a-4e89-bd1f-6824fbcfb2b6",
    "cluster_name": "mrs_noauth_autotest_do_not_del",
    "create_time": 1715671906000,
    "create_user": "ei_dayu_xxxx_01",
    "database_name": "dls_property",
    "datasource_type": "HIVE",
    "id": "fa0adf4b660fe1a61abc7f990974537e",
    "name": "test100",
    "sync_msg": "",
    "sync_status": "NOT_SYNC",
    "sync_time": 1715756630000,
    "table_name": "test100",
    "update_time": 1715915649000,
    "update_user": "ei_dayu_xxxx_01",
    "user_groups": "yrc,fanxianggen_test",
    "users": "autotest_tics,cdos_test"
  } ],
  "total": 1
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ListSecurityDynamicMaskingPoliciesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSecurityDynamicMaskingPoliciesRequest request = new
        ListSecurityDynamicMaskingPoliciesRequest();
        try {
            ListSecurityDynamicMaskingPoliciesResponse response =
            client.listSecurityDynamicMaskingPolicies(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DataArtsStudioClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListSecurityDynamicMaskingPoliciesRequest()
    response = client.list_security_dynamic_masking_policies(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecurityDynamicMaskingPoliciesRequest{}
    response, err := client.ListSecurityDynamicMaskingPolicies(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```



## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.14.2 创建数据脱敏策略

### 功能介绍

创建动态数据脱敏策略。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/masking/dynamic/policies

表 12-345 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-346 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-347 请求 Body 参数

| 参数              | 是否必选 | 参数类型                                                        | 描述                                                                                                    |
|-----------------|------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| name            | 是    | String                                                      | 策略名称。英文和汉字开头, 支持英文、汉字、数字、下划线, 2-64字符。                                                                 |
| datasource_type | 是    | String                                                      | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| cluster_id      | 是    | String                                                      | 集群id。请于集群管理页面查看集群ID信息。当数据源类型为DLI时, 该参数需要填写为DLI。                                                       |
| cluster_name    | 是    | String                                                      | 集群名称。请于集群管理页面查看集群名称信息。当数据源类型为DLI时, 该参数需要填写为DLI。                                                       |
| database_name   | 是    | String                                                      | 数据库名称。获取方法请参见获取数据源中的表。                                                                                |
| table_id        | 否    | String                                                      | 数据表id, 获取方法请参见获取数据源中的表。                                                                               |
| table_name      | 是    | String                                                      | 数据表名称, 获取方法请参见获取数据源中的表。                                                                               |
| user_groups     | 否    | String                                                      | 用户组列表, 用户组名称逗号分隔 (非必填项, 但用户、用户组必须二选其一进行配置)。例如:<br>"userGroup1,userGroup2"。                            |
| users           | 否    | String                                                      | 用户列表, 用户名称逗号分隔 (非必填项, 但用户、用户组必须二选其一进行配置), 例如:<br>"user1,user2"。                                       |
| conn_name       | 是    | String                                                      | 数据连接名称, 获取方法请参见 <a href="#">查询数据连接列表</a> 。                                                            |
| conn_id         | 是    | String                                                      | 数据连接id, 获取方法请参见 <a href="#">查询数据连接列表</a> 。                                                            |
| schema_name     | 否    | String                                                      | DWS数据源的模式名称。                                                                                          |
| policy_list     | 是    | Array of <a href="#">DynamicMaskingPolicyCreate</a> objects | 动态数据脱敏策略列表。                                                                                           |

表 12-348 DynamicMaskingPolicyCreate

| 参数               | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| column_name      | 是    | String | 数据表中的字段名称。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| column_type      | 是    | String | 数据表中字段的数据类型。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| algorithm_type   | 否    | String | <p>具体动态脱敏规则参数介绍请参见DataArts Studio用户指南手册中“管理动态脱敏策略”章节。</p> <p>HIVE数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> <p>DWS数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• DWS_ALL_MASK 全掩码</li> <li>• DWS_BACK_KEEP 保留后4位，其余脱敏为*</li> <li>• DWS_FRONT_KEEP 保留前2位，其余脱敏为*</li> <li>• DWS_SELF_CONFIG 需要输入开始位置、结束位置、脱敏字符传入detail结构体中，例如{"start": 1, "end": 2, "string_target": "*"}</li> </ul> <p>DLI数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> |
| algorithm_detail | 否    | String | 动态脱敏策略算法详情。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| 参数                   | 是否必选 | 参数类型                      | 描述         |
|----------------------|------|---------------------------|------------|
| algorithm_detail_dto | 否    | AlgorithmDetailDTO object | 动态脱敏策略算法详情 |

表 12-349 AlgorithmDetailDTO

| 参数            | 是否必选 | 参数类型    | 描述                    |
|---------------|------|---------|-----------------------|
| start         | 否    | Integer | 开始位数，该值需要大于0且小于end值。  |
| end           | 否    | Integer | 结束位数，该值需要大于或等于start值。 |
| int_target    | 否    | Integer | 数值类型。                 |
| string_target | 否    | String  | 字符串类型。可输入参数为"*"或者"#". |

## 响应参数

状态码： 200

表 12-350 响应 Body 参数

| 参数              | 参数类型   | 描述                                                                                                    |
|-----------------|--------|-------------------------------------------------------------------------------------------------------|
| id              | String | 策略id。                                                                                                 |
| name            | String | 策略名称。英文和汉字开头, 支持英文、汉字、数字、下划线, 2-64字符。                                                                 |
| datasource_type | String | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| cluster_id      | String | 集群id。请于集群管理页面查看集群ID信息。当数据源类型为DLI时，该参数需要填写为DLI。                                                        |
| cluster_name    | String | 集群名称。请于集群管理页面查看集群名称信息。当数据源类型为DLI时，该参数需要填写为DLI。                                                        |
| database_name   | String | 数据库名称。获取方法请参见获取数据源中的表。                                                                                |
| table_id        | String | 数据表id，获取方法请参见获取数据源中的表。                                                                                |

| 参数          | 参数类型                                                  | 描述                                                                                                                                                                                                                                                                                                                        |
|-------------|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| table_name  | String                                                | 数据表名称, 获取方法请参见获取数据源中的表。                                                                                                                                                                                                                                                                                                   |
| user_groups | String                                                | 用户组列表, 用户组名称逗号分隔 (非必填项, 但用户、用户组必须二选其一进行配置)。例如: "userGroup1,userGroup2"。                                                                                                                                                                                                                                                   |
| users       | String                                                | 用户列表, 用户名称逗号分隔 (非必填项, 但用户、用户组必须二选其一进行配置), 例如: "user1,user2"。                                                                                                                                                                                                                                                              |
| conn_name   | String                                                | 数据连接名称, 获取方法请参见 <a href="#">查询数据连接列表</a> 。                                                                                                                                                                                                                                                                                |
| conn_id     | String                                                | 数据连接id, 获取方法请参见 <a href="#">查询数据连接列表</a> 。                                                                                                                                                                                                                                                                                |
| sync_status | String                                                | 同步状态: <ul style="list-style-type: none"> <li>• UNKNOWN 未知状态</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> <li>• SYNC_PARTIAL_FAIL 存在失败</li> <li>• DELETE_FAIL 删除失败</li> <li>• DELETING 删除中</li> <li>• UPDATING 更新中</li> <li>• DATA_UPDATED 数据存在更新</li> </ul> |
| sync_msg    | String                                                | 策略同步信息。                                                                                                                                                                                                                                                                                                                   |
| sync_log    | String                                                | 同步运行日志, 格式为 字段同步信息+换行符。                                                                                                                                                                                                                                                                                                   |
| create_time | Long                                                  | 策略创建时间。                                                                                                                                                                                                                                                                                                                   |
| create_user | String                                                | 策略创建者。                                                                                                                                                                                                                                                                                                                    |
| update_time | Long                                                  | 策略更新时间。                                                                                                                                                                                                                                                                                                                   |
| update_user | String                                                | 策略更新者。                                                                                                                                                                                                                                                                                                                    |
| schema_name | String                                                | DWS数据源的模式名称。                                                                                                                                                                                                                                                                                                              |
| policy_list | Array of <a href="#">DynamicMaskingPolicy</a> objects | 动态数据脱敏策略列表。                                                                                                                                                                                                                                                                                                               |

表 12-351 DynamicMaskingPolicy

| 参数             | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id             | String | 字段脱敏策略id。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| policy_set_id  | String | 动态脱敏策略id。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| column_name    | String | 数据表中的字段名称。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| column_type    | String | 数据表中字段的数据类型。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| algorithm_type | String | <p>动态脱敏规则参数介绍，具体请参考用户指南手册中的“动态脱敏规则介绍”章节。</p> <p>HIVE数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> <p>DWS数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• DWS_ALL_MASK 全掩码</li> <li>• DWS_BACK_KEEP 保留后4位，其余脱敏为*</li> <li>• DWS_FRONT_KEEP 保留前2位，其余脱敏为*</li> <li>• DWS_SELF_CONFIG 需要输入开始位置、结束位置、脱敏字符传入detail结构体中，例如 {"start": 1, "end": 2, "string_target": "*"}</li> </ul> <p>DLI数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> |

| 参数                   | 参数类型                      | 描述                                                                                                                                                                                                                                                                                                                        |
|----------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sync_status          | String                    | 同步状态： <ul style="list-style-type: none"> <li>• UNKNOWN 未知状态</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> <li>• SYNC_PARTIAL_FAIL 存在失败</li> <li>• DELETE_FAIL 删除失败</li> <li>• DELETING 删除中</li> <li>• UPDATING 更新中</li> <li>• DATA_UPDATED 数据存在更新</li> </ul> |
| algorithm_detail     | String                    | 动态脱敏策略算法详情。                                                                                                                                                                                                                                                                                                               |
| algorithm_detail_dto | AlgorithmDetailDTO object | 动态脱敏策略算法详情                                                                                                                                                                                                                                                                                                                |

表 12-352 AlgorithmDetailDTO

| 参数            | 参数类型    | 描述                    |
|---------------|---------|-----------------------|
| start         | Integer | 开始位数，该值需要大于0且小于end值。  |
| end           | Integer | 结束位数，该值需要大于或等于start值。 |
| int_target    | Integer | 数值类型。                 |
| string_target | String  | 字符串类型。可输入参数为"*"或者"#". |

状态码： 400

表 12-353 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

- 创建HIVE数据源的动态脱敏策略

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/masking/dynamic/policies
```

```
{
  "name": "OpenAPI_Hive_test",
  "datasource_type": "HIVE",
  "cluster_id": "dc425074-26b3-479c-9e2f-b103c0cdd90f",
  "cluster_name": "mrs_3x_xxxx_do_not_del",
  "database_name": "bigdatatest",
  "table_name": "aaaa",
  "table_id": "NativeTable-39dc19fb17034ab39e46ebe0420c6202-bigdatatest-aaaa",
  "user_groups": "DataArts User",
  "users": "",
  "conn_name": "hive_3x_0330",
  "conn_id": "39dc19fb17034ab39e46ebe0420c6202",
  "policy_list": [ {
    "column_name": "aa",
    "column_type": "string",
    "algorithm_type": "MASK_SHOW_LAST_4"
  } ]
}
```

- 创建DWS数据源的动态脱敏策略

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/masking/dynamic/policies
```

```
{
  "name": "OpenAPI_DWS_test",
  "datasource_type": "DWS",
  "cluster_id": "c94a9133-ef65-47ad-9483-bfa8c5eaa101",
  "cluster_name": "dws_ssl_4autotest_nomodify",
  "database_name": "gaussdb",
  "table_name": "all_column",
  "table_id": "NativeTable-e2f6c44100654778aee017f8ad9e3ac9-gaussdb-public-all_column",
  "user_groups": "DataArts_PDP5_NoPrivilege",
  "users": "autotest_tics",
  "conn_name": "dws_0430",
  "conn_id": "e2f6c44100654778aee017f8ad9e3ac9",
  "policy_list": [ {
    "column_name": "avarchar",
    "column_type": "varchar",
    "algorithm_type": "DWS_ALL_MASK",
    "algorithm_detail_dto": null
  } ],
  "schema_name": "public"
}
```

- 创建DLI数据源的动态脱敏策略

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/masking/dynamic/policies
```

```
{
  "name": "OpenAPI_DLI_test",
  "datasource_type": "DLI",
  "cluster_id": "DLI",
  "cluster_name": "DLI",
  "database_name": "bigdatatest",
  "table_name": "dli_string_auto_20231116",
  "table_id": "NativeTable-619d52a0e6954aa68844f5f010e06ef8-bigdatatest-dli_string_auto_20231116",
  "user_groups": "",
  "users": "",
  "conn_name": "xu_dli_link",
  "conn_id": "619d52a0e6954aa68844f5f010e06ef8",
  "policy_list": [ {
    "column_name": "varchar1",
    "column_type": "string",
    "algorithm_type": "MASK_NULL"
  } ]
}
```



## 响应示例

状态码： 200

OK

```
{
  "cluster_id": "dc425074-26b3-479c-9e2f-b103c0cdd90f",
  "cluster_name": "mrs_3x_autotest_do_not_del",
  "conn_id": "ae55cb5c44be4119b8958a3ba5d9a71f",
  "conn_name": "hive_xxxx_0520",
  "create_time": 1716195352790,
  "create_user": "ei_dayu_xxxx_01",
  "database_name": "bigdatatest",
  "datasource_type": "HIVE",
  "id": "e55577e76cb77f87cec2099b8074a88d",
  "name": "xxxx_unique_test",
  "policy_list": [ {
    "algorithm_detail": null,
    "algorithm_detail_dto": null,
    "algorithm_type": "MASK_SHOW_LAST_4",
    "associated_id": null,
    "associated_policy_name": null,
    "column_name": "name",
    "column_type": "string",
    "id": "607218bdfdf8452cd0abe9354639c472",
    "policy_set_id": "e55577e76cb77f87cec2099b8074a88d",
    "sync_msg": null,
    "sync_status": "NOT_SYNC",
    "sync_time": null
  }, {
    "algorithm_detail": null,
    "algorithm_detail_dto": null,
    "algorithm_type": "MASK_SHOW_FIRST_4",
    "associated_id": null,
    "associated_policy_name": null,
    "column_name": "a1",
    "column_type": "string",
    "id": "95cefff4b29e1db7fedb9b772739d7a0",
    "policy_set_id": "e55577e76cb77f87cec2099b8074a88d",
    "sync_msg": null,
    "sync_status": "NOT_SYNC",
    "sync_time": null
  } ],
  "schema_id": null,
  "schema_name": null,
  "sync_log": null,
  "sync_msg": null,
  "sync_status": "NOT_SYNC",
  "table_id": "NativeTable-ae55cb5c44be4119b8958a3ba5d9a71f-bigdatatest-dws_gglllsss",
  "table_name": "dws_gglllsss",
  "update_time": 1716195352790,
  "update_user": "ei_dayu_xxxx_01",
  "user_groups": "DataArts User",
  "users": "user1,user2,user3,user4"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 创建HIVE数据源的动态脱敏策略

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateSecurityDynamicMaskingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateSecurityDynamicMaskingPolicyRequest request = new
        CreateSecurityDynamicMaskingPolicyRequest();
        DynamicMaskingPolicyCreateDTO body = new DynamicMaskingPolicyCreateDTO();
        List<DynamicMaskingPolicyCreate> listbodyPolicyList = new ArrayList<>();
        listbodyPolicyList.add(
            new DynamicMaskingPolicyCreate()
                .withColumnName("aa")
                .withColumnType("string")
                .withAlgorithmType("MASK_SHOW_LAST_4")
        );
        body.withPolicyList(listbodyPolicyList);
        body.withConnId("39dc19fb17034ab39e46ebe0420c6202");
        body.withConnName("hive_3x_0330");
        body.withUsers("");
        body.withUserGroups("DataArts User");
        body.withTableName("aaaa");
        body.withTableId("NativeTable-39dc19fb17034ab39e46ebe0420c6202-bigdatatest-aaaa");
        body.withDatabaseName("bigdatatest");
        body.withClusterName("mrs_3x_xxxx_do_not_del");
        body.withClusterId("dc425074-26b3-479c-9e2f-b103c0cdd90f");

        body.withDatasourceType(DynamicMaskingPolicyCreateDTO.DatasourceTypeEnum.fromValue("HIVE"));
        ;
        body.withName("OpenAPI_Hive_test");
        request.withBody(body);
        try {
            CreateSecurityDynamicMaskingPolicyResponse response =
            client.createSecurityDynamicMaskingPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
        }
    }
}
```

```

        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}

```

- 创建DWS数据源的动态脱敏策略

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateSecurityDynamicMaskingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateSecurityDynamicMaskingPolicyRequest request = new
        CreateSecurityDynamicMaskingPolicyRequest();
        DynamicMaskingPolicyCreateDTO body = new DynamicMaskingPolicyCreateDTO();
        List<DynamicMaskingPolicyCreate> listbodyPolicyList = new ArrayList<>();
        listbodyPolicyList.add(
            new DynamicMaskingPolicyCreate()
                .withColumnName("avarchar")
                .withColumnType("varchar")
                .withAlgorithmType("DWS_ALL_MASK")
        );
        body.withPolicyList(listbodyPolicyList);
        body.withSchemaName("public");
        body.withConnId("e2f6c44100654778aee017f8ad9e3ac9");
        body.withConnName("dws_0430");
        body.withUsers("autotest_tics");
        body.withUserGroups("DataArts_PDP5_NoPrivilleage");
        body.withTableName("all_column");
        body.withTableId("NativeTable-e2f6c44100654778aee017f8ad9e3ac9-gaussdb-public-
all_column");
        body.withDatabaseName("gaussdb");
        body.withClusterName("dws_ssl_4autotest_nomodify");
        body.withClusterId("c94a9133-ef65-47ad-9483-bfa8c5eaa101");

        body.withDatasourceType(DynamicMaskingPolicyCreateDTO.DatasourceTypeEnum.fromValue("DWS"));
        ;
        body.withName("OpenAPI_DWS_test");
    }
}

```

```

request.withBody(body);
try {
    CreateSecurityDynamicMaskingPolicyResponse response =
client.createSecurityDynamicMaskingPolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

- 创建DLI数据源的动态脱敏策略

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateSecurityDynamicMaskingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateSecurityDynamicMaskingPolicyRequest request = new
CreateSecurityDynamicMaskingPolicyRequest();
        DynamicMaskingPolicyCreateDTO body = new DynamicMaskingPolicyCreateDTO();
        List<DynamicMaskingPolicyCreate> listbodyPolicyList = new ArrayList<>();
        listbodyPolicyList.add(
            new DynamicMaskingPolicyCreate()
                .withColumnName("varchar1")
                .withColumnType("string")
                .withAlgorithmType("MASK_NULL")
        );
        body.withPolicyList(listbodyPolicyList);
        body.withConnId("619d52a0e6954aa68844f5f010e06ef8");
        body.withConnName("xu_dli_link");
        body.withUsers("");
    }
}

```

```
        body.withUserGroups("");
        body.withTableName("dli_string_auto_20231116");
        body.withTableId("NativeTable-619d52a0e6954aa68844f5f010e06ef8-bigdatatest-
dli_string_auto_20231116");
        body.withDatabaseName("bigdatatest");
        body.withClusterName("DLI");
        body.withClusterId("DLI");

body.withDatasourceType(DynamicMaskingPolicyCreateDTO.DatasourceTypeEnum.fromValue("DLI"));
body.withName("OpenAPI_DLI_test");
request.withBody(body);
try {
    CreateSecurityDynamicMaskingPolicyResponse response =
client.createSecurityDynamicMaskingPolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

- 创建HIVE数据源的动态脱敏策略

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecurityDynamicMaskingPolicyRequest()
        listPolicyListbody = [
            DynamicMaskingPolicyCreate(
                column_name="aa",
                column_type="string",
                algorithm_type="MASK_SHOW_LAST_4"
            )
        ]
        request.body = DynamicMaskingPolicyCreateDTO(
            policy_list=listPolicyListbody,
```

```
conn_id="39dc19fb17034ab39e46ebe0420c6202",
conn_name="hive_3x_0330",
users="",
user_groups="DataArts User",
table_name="aaaa",
table_id="NativeTable-39dc19fb17034ab39e46ebe0420c6202-bigdatatest-aaaa",
database_name="bigdatatest",
cluster_name="mrs_3x_xxxx_do_not_del",
cluster_id="dc425074-26b3-479c-9e2f-b103c0cdd90f",
datasource_type="HIVE",
name="OpenAPI_Hive_test"
)
response = client.create_security_dynamic_masking_policy(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 创建DWS数据源的动态脱敏策略

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecurityDynamicMaskingPolicyRequest()
        listPolicyListbody = [
            DynamicMaskingPolicyCreate(
                column_name="avarchar",
                column_type="varchar",
                algorithm_type="DWS_ALL_MASK"
            )
        ]
        request.body = DynamicMaskingPolicyCreateDTO(
            policy_list=listPolicyListbody,
            schema_name="public",
            conn_id="e2f6c44100654778aee017f8ad9e3ac9",
            conn_name="dws_0430",
            users="autotest_tics",
            user_groups="DataArts_PDP5_NoPrivilege",
            table_name="all_column",
            table_id="NativeTable-e2f6c44100654778aee017f8ad9e3ac9-gaussdb-public-all_column",
            database_name="gaussdb",
            cluster_name="dws_ssl_4autotest_nomodify",
            cluster_id="c94a9133-ef65-47ad-9483-bfa8c5eaa101",
            datasource_type="DWS",
            name="OpenAPI_DWS_test"
```

```
)  
response = client.create_security_dynamic_masking_policy(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

- 创建DLI数据源的动态脱敏策略

```
# coding: utf-8
```

```
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdataartsstudio.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    # environment variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before  
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
    # environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DataArtsStudioClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateSecurityDynamicMaskingPolicyRequest()  
        listPolicyListbody = [  
            DynamicMaskingPolicyCreate(  
                column_name="varchar1",  
                column_type="string",  
                algorithm_type="MASK_NULL"  
            )  
        ]  
        request.body = DynamicMaskingPolicyCreateDTO(  
            policy_list=listPolicyListbody,  
            conn_id="619d52a0e6954aa68844f5f010e06ef8",  
            conn_name="xu_dli_link",  
            users="",  
            user_groups="",  
            table_name="dli_string_auto_20231116",  
            table_id="NativeTable-619d52a0e6954aa68844f5f010e06ef8-bigdatatest-  
dli_string_auto_20231116",  
            database_name="bigdatatest",  
            cluster_name="DLI",  
            cluster_id="DLI",  
            datasource_type="DLI",  
            name="OpenAPI_DLI_test"  
        )  
        response = client.create_security_dynamic_masking_policy(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

- 创建HIVE数据源的动态脱敏策略

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityDynamicMaskingPolicyRequest{
        algorithmTypePolicyList:= "MASK_SHOW_LAST_4"
        var listPolicyListbody = []model.DynamicMaskingPolicyCreate{
            {
                ColumnName: "aa",
                ColumnType: "string",
                AlgorithmType: &algorithmTypePolicyList,
            },
        },
        usersDynamicMaskingPolicyCreateDto:= ""
        userGroupsDynamicMaskingPolicyCreateDto:= "DataArts User"
        tableIdDynamicMaskingPolicyCreateDto:= "NativeTable-39dc19fb17034ab39e46ebe0420c6202-
        bigdatatest-aaaa"
        request.Body = &model.DynamicMaskingPolicyCreateDto{
            PolicyList: listPolicyListbody,
            ConnId: "39dc19fb17034ab39e46ebe0420c6202",
            ConnName: "hive_3x_0330",
            Users: &usersDynamicMaskingPolicyCreateDto,
            UserGroups: &userGroupsDynamicMaskingPolicyCreateDto,
            TableName: "aaaa",
            TableId: &tableIdDynamicMaskingPolicyCreateDto,
            DatabaseName: "bigdatatest",
            ClusterName: "mrs_3x_xxxx_do_not_del",
            ClusterId: "dc425074-26b3-479c-9e2f-b103c0cdd90f",
            DataSourceType: model.GetDynamicMaskingPolicyCreateDtoDataSourceTypeEnum().HIVE,
            Name: "OpenAPI_Hive_test",
        }
        response, err := client.CreateSecurityDynamicMaskingPolicy(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```



- 创建DWS数据源的动态脱敏策略

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityDynamicMaskingPolicyRequest{
        algorithmTypePolicyList:= "DWS_ALL_MASK"
        var listPolicyListbody = []model.DynamicMaskingPolicyCreate{
            {
                ColumnName: "avarchar",
                ColumnType: "varchar",
                AlgorithmType: &algorithmTypePolicyList,
            },
        },
        schemaNameDynamicMaskingPolicyCreateDto:= "public"
        usersDynamicMaskingPolicyCreateDto:= "autotest_tics"
        userGroupsDynamicMaskingPolicyCreateDto:= "DataArts_PDP5_NoPrivilleage"
        tableIdDynamicMaskingPolicyCreateDto:= "NativeTable-e2f6c44100654778aee017f8ad9e3ac9-
        gaussdb-public-all_column"
        request.Body = &model.DynamicMaskingPolicyCreateDto{
            PolicyList: listPolicyListbody,
            SchemaName: &schemaNameDynamicMaskingPolicyCreateDto,
            ConnId: "e2f6c44100654778aee017f8ad9e3ac9",
            ConnName: "dws_0430",
            Users: &usersDynamicMaskingPolicyCreateDto,
            UserGroups: &userGroupsDynamicMaskingPolicyCreateDto,
            TableName: "all_column",
            TableId: &tableIdDynamicMaskingPolicyCreateDto,
            DatabaseName: "gaussdb",
            ClusterName: "dws_ssl_4autotest_nomodify",
            ClusterId: "c94a9133-ef65-47ad-9483-bfa8c5eaa101",
            DatasourceType: model.GetDynamicMaskingPolicyCreateDtoDatasourceTypeEnum().DWS,
            Name: "OpenAPI_DWS_test",
        }
    }
    response, err := client.CreateSecurityDynamicMaskingPolicy(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
```

```
    fmt.Println(err)
  }
}
```

- 创建DLI数据源的动态脱敏策略

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecurityDynamicMaskingPolicyRequest{
        algorithmTypePolicyList:= "MASK_NULL"
        var listPolicyListbody = []model.DynamicMaskingPolicyCreate{
            {
                ColumnName: "varchar1",
                ColumnType: "string",
                AlgorithmType: &algorithmTypePolicyList,
            },
        },
    }
    usersDynamicMaskingPolicyCreateDto:= ""
    userGroupsDynamicMaskingPolicyCreateDto:= ""
    tableIdDynamicMaskingPolicyCreateDto:= "NativeTable-619d52a0e6954aa68844f5f010e06ef8-
bigdatatest-dli_string_auto_20231116"
    request.Body = &model.DynamicMaskingPolicyCreateDto{
        PolicyList: listPolicyListbody,
        ConnId: "619d52a0e6954aa68844f5f010e06ef8",
        ConnName: "xu_dli_link",
        Users: &usersDynamicMaskingPolicyCreateDto,
        UserGroups: &userGroupsDynamicMaskingPolicyCreateDto,
        TableName: "dli_string_auto_20231116",
        TableId: &tableIdDynamicMaskingPolicyCreateDto,
        DatabaseName: "bigdatatest",
        ClusterName: "DLI",
        ClusterId: "DLI",
        DatasourceType: model.GetDynamicMaskingPolicyCreateDtoDatasourceTypeEnum().DLI,
        Name: "OpenAPI_DLI_test",
    }
    response, err := client.CreateSecurityDynamicMaskingPolicy(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.14.3 查询某个脱敏策略的详细信息

### 功能介绍

查询某个脱敏策略的详细信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/masking/dynamic/policies/{id}

表 12-354 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 动态脱敏策略id。                                |

### 请求参数

表 12-355 请求 Header 参数

| 参数        | 是否必选 | 参数类型   | 描述                                           |
|-----------|------|--------|----------------------------------------------|
| workspace | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。 |

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 200

表 12-356 响应 Body 参数

| 参数              | 参数类型   | 描述                                                                                                    |
|-----------------|--------|-------------------------------------------------------------------------------------------------------|
| id              | String | 策略id。                                                                                                 |
| name            | String | 策略名称。英文和汉字开头, 支持英文、汉字、数字、下划线, 2-64字符。                                                                 |
| datasource_type | String | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| cluster_id      | String | 集群id。请于集群管理页面查看集群ID信息。当数据源类型为DLI时，该参数需要填写为DLI。                                                        |
| cluster_name    | String | 集群名称。请于集群管理页面查看集群名称信息。当数据源类型为DLI时，该参数需要填写为DLI。                                                        |
| database_name   | String | 数据库名称。获取方法请参见获取数据源中的表。                                                                                |
| table_id        | String | 数据表id，获取方法请参见获取数据源中的表。                                                                                |
| table_name      | String | 数据表名称, 获取方法请参见获取数据源中的表。                                                                               |
| user_groups     | String | 用户组列表，用户组名称逗号分隔（非必填项，但用户、用户组必须二选其一进行配置）。例如："userGroup1,userGroup2"。                                   |
| users           | String | 用户列表，用户名称逗号分隔（非必填项，但用户、用户组必须二选其一进行配置），例如："user1,user2"。                                               |
| conn_name       | String | 数据连接名称，获取方法请参见 <a href="#">查询数据连接列表</a> 。                                                             |
| conn_id         | String | 数据连接id，获取方法请参见 <a href="#">查询数据连接列表</a> 。                                                             |

| 参数          | 参数类型                                         | 描述                                                                                                                                                                                                                                                                                                                        |
|-------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sync_status | String                                       | 同步状态： <ul style="list-style-type: none"> <li>• UNKNOWN 未知状态</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> <li>• SYNC_PARTIAL_FAIL 存在失败</li> <li>• DELETE_FAIL 删除失败</li> <li>• DELETING 删除中</li> <li>• UPDATING 更新中</li> <li>• DATA_UPDATED 数据存在更新</li> </ul> |
| sync_msg    | String                                       | 策略同步信息。                                                                                                                                                                                                                                                                                                                   |
| sync_log    | String                                       | 同步运行日志, 格式为 字段同步信息+换行符。                                                                                                                                                                                                                                                                                                   |
| create_time | Long                                         | 策略创建时间。                                                                                                                                                                                                                                                                                                                   |
| create_user | String                                       | 策略创建者。                                                                                                                                                                                                                                                                                                                    |
| update_time | Long                                         | 策略更新时间。                                                                                                                                                                                                                                                                                                                   |
| update_user | String                                       | 策略更新者。                                                                                                                                                                                                                                                                                                                    |
| schema_name | String                                       | DWS数据源的模式名称。                                                                                                                                                                                                                                                                                                              |
| policy_list | Array of <b>DynamicMaskingPolicy</b> objects | 动态数据脱敏策略列表。                                                                                                                                                                                                                                                                                                               |

表 12-357 DynamicMaskingPolicy

| 参数            | 参数类型   | 描述           |
|---------------|--------|--------------|
| id            | String | 字段脱敏策略id。    |
| policy_set_id | String | 动态脱敏策略id。    |
| column_name   | String | 数据表中的字段名称。   |
| column_type   | String | 数据表中字段的数据类型。 |

| 参数               | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| algorithm_type   | String | <p>动态脱敏规则参数介绍，具体请参考用户指南手册中的“动态脱敏规则介绍”章节。</p> <p>HIVE数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> <p>DWS数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• DWS_ALL_MASK 全掩码</li> <li>• DWS_BACK_KEEP 保留后4位，其余脱敏为*</li> <li>• DWS_FRONT_KEEP 保留前2位，其余脱敏为*</li> <li>• DWS_SELF_CONFIG 需要输入开始位置、结束位置、脱敏字符传入detail结构体中，例如 {"start": 1, "end": 2, "string_target": "*"}</li> </ul> <p>DLI数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> |
| sync_status      | String | <p>同步状态：</p> <ul style="list-style-type: none"> <li>• UNKNOWN 未知状态</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> <li>• SYNC_PARTIAL_FAIL 存在失败</li> <li>• DELETE_FAIL 删除失败</li> <li>• DELETING 删除中</li> <li>• UPDATING 更新中</li> <li>• DATA_UPDATED 数据存在更新</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| algorithm_detail | String | 动态脱敏策略算法详情。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

| 参数                   | 参数类型                                | 描述         |
|----------------------|-------------------------------------|------------|
| algorithm_detail_dto | <b>AlgorithmDetailDTO</b><br>object | 动态脱敏策略算法详情 |

表 12-358 AlgorithmDetailDTO

| 参数            | 参数类型    | 描述                    |
|---------------|---------|-----------------------|
| start         | Integer | 开始位数，该值需要大于0且小于end值。  |
| end           | Integer | 结束位数，该值需要大于或等于start值。 |
| int_target    | Integer | 数值类型。                 |
| string_target | String  | 字符串类型。可输入参数为"*"或者"#". |

状态码： 400

表 12-359 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

无

## 响应示例

状态码： 200

动态数据脱敏策略

```
{
  "cluster_id": "dc425074-26b3-479c-9e2f-b103c0cdd90f",
  "cluster_name": "mrs_3x_xxxx_do_not_del",
  "conn_id": "ae55cb5c44be4119b8958a3ba5d9a71f",
  "conn_name": "hive_xxxx_0520",
  "create_time": 1716195990000,
  "create_user": "ei_dayu_xxxx_01",
  "database_name": "bigdatatest",
  "datasource_type": "HIVE",
  "id": "7ea6b287dfd34b81c5eb627de2fd3401",
  "name": "xxxx",
  "policy_list": [ {
    "algorithm_detail": null,
    "algorithm_detail_dto": null,
    "algorithm_type": "MASK_HASH",
```

```
"associated_id" : null,
"associated_policy_name" : null,
"column_name" : "rr",
"column_type" : "string",
"id" : "3ced68b49ce558507af65207d880e51f",
"policy_set_id" : "7ea6b287dfd34b81c5eb627de2fd3401",
"sync_msg" : null,
"sync_status" : "NOT_SYNC",
"sync_time" : null
}],
"schema_id" : null,
"schema_name" : null,
"sync_log" : "",
"sync_msg" : null,
"sync_status" : "NOT_SYNC",
"table_id" : "NativeTable-ae55cb5c44be4119b8958a3ba5d9a71f-bigdatatest-chuan",
"table_name" : "xxxx",
"update_time" : 1716195990000,
"update_user" : "ei_dayu_xxxx_01",
"user_groups" : "xxxx",
"users" : ""
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

public class ShowSecurityDynamicMaskingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowSecurityDynamicMaskingPolicyRequest request = new
        ShowSecurityDynamicMaskingPolicyRequest();
        request.withId("{id}");
        try {
            ShowSecurityDynamicMaskingPolicyResponse response =
            client.showSecurityDynamicMaskingPolicy(request);
        }
    }
}
```



```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecurityDynamicMaskingPolicyRequest()
        request.id = "{id}"
        response = client.show_security_dynamic_masking_policy(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dataartsstudio.NewDataArtsStudioClient(
    dataartsstudio.DataArtsStudioClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowSecurityDynamicMaskingPolicyRequest{}
request.Id = "{id}"
response, err := client.ShowSecurityDynamicMaskingPolicy(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | 动态数据脱敏策略    |
| 400 | Bad Request |

## 12.14.4 更新动态数据脱敏策略

### 功能介绍

更新动态数据脱敏策略。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/{project\_id}/security/masking/dynamic/policies/{id}

表 12-360 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 脱敏策略id                                   |

## 请求参数

表 12-361 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-362 请求 Body 参数

| 参数              | 是否必选 | 参数类型   | 描述                                                                                                    |
|-----------------|------|--------|-------------------------------------------------------------------------------------------------------|
| name            | 是    | String | 策略名称。英文和汉字开头，支持英文、汉字、数字、下划线，2-64字符。                                                                   |
| datasource_type | 是    | String | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| cluster_id      | 是    | String | 集群id。请于集群管理页面查看集群ID信息。当数据源类型为DLI时，该参数需要填写为DLI。                                                        |
| cluster_name    | 是    | String | 集群名称。请于集群管理页面查看集群名称信息。当数据源类型为DLI时，该参数需要填写为DLI。                                                        |
| database_name   | 是    | String | 数据库名称。获取方法请参见获取数据源中的表。                                                                                |
| table_id        | 否    | String | 数据表id，获取方法请参见获取数据源中的表。                                                                                |

| 参数          | 是否必选 | 参数类型                                               | 描述                                                                         |
|-------------|------|----------------------------------------------------|----------------------------------------------------------------------------|
| table_name  | 是    | String                                             | 数据表名称, 获取方法请参见获取数据源中的表。                                                    |
| user_groups | 否    | String                                             | 用户组列表, 用户组名称逗号分隔 (非必填项, 但用户、用户组必须二选其一进行配置)。例如:<br>"userGroup1,userGroup2"。 |
| users       | 否    | String                                             | 用户列表, 用户名称逗号分隔 (非必填项, 但用户、用户组必须二选其一进行配置), 例如:<br>"user1,user2"。            |
| conn_name   | 是    | String                                             | 数据连接名称, 获取方法请参见 <a href="#">查询数据连接列表</a> 。                                 |
| conn_id     | 是    | String                                             | 数据连接id, 获取方法请参见 <a href="#">查询数据连接列表</a> 。                                 |
| schema_name | 否    | String                                             | DWS数据源的模式名称。                                                               |
| policy_list | 是    | Array of <b>DynamicMaskingPolicyUpdate</b> objects | 动态数据脱敏策略列表。                                                                |

表 12-363 DynamicMaskingPolicyUpdate

| 参数          | 是否必选 | 参数类型   | 描述           |
|-------------|------|--------|--------------|
| column_name | 是    | String | 数据表中的字段名称。   |
| column_type | 是    | String | 数据表中字段的数据类型。 |

| 参数                   | 是否必选 | 参数类型                                | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| algorithm_type       | 否    | String                              | <p>动态脱敏规则参数介绍，具体请参考用户指南手册中的“动态脱敏规则介绍”章节。</p> <p>HIVE数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> <p>DWS数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• DWS_ALL_MASK 全掩码</li> <li>• DWS_BACK_KEEP 保留后4位，其余脱敏为*</li> <li>• DWS_FRONT_KEEP 保留前2位，其余脱敏为*</li> <li>• DWS_SELF_CONFIG 需要输入开始位置、结束位置、脱敏字符传入detail结构体中，例如{"start": 1, "end": 2, "string_target": "*"}</li> </ul> <p>DLI数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> |
| algorithm_detail     | 否    | String                              | 动态脱敏策略算法详情。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| algorithm_detail_dto | 否    | <b>AlgorithmDetailDTO</b><br>object | 动态脱敏策略算法详情                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

表 12-364 AlgorithmDetailDTO

| 参数            | 是否必选 | 参数类型    | 描述                    |
|---------------|------|---------|-----------------------|
| start         | 否    | Integer | 开始位数，该值需要大于0且小于end值。  |
| end           | 否    | Integer | 结束位数，该值需要大于或等于start值。 |
| int_target    | 否    | Integer | 数值类型。                 |
| string_target | 否    | String  | 字符串类型。可输入参数为"*"或者"#". |

## 响应参数

状态码： 200

表 12-365 响应 Body 参数

| 参数              | 参数类型   | 描述                                                                                                    |
|-----------------|--------|-------------------------------------------------------------------------------------------------------|
| id              | String | 策略id。                                                                                                 |
| name            | String | 策略名称。英文和汉字开头, 支持英文、汉字、数字、下划线, 2-64字符。                                                                 |
| datasource_type | String | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| cluster_id      | String | 集群id。请于集群管理页面查看集群ID信息。当数据源类型为DLI时，该参数需要填写为DLI。                                                        |
| cluster_name    | String | 集群名称。请于集群管理页面查看集群名称信息。当数据源类型为DLI时，该参数需要填写为DLI。                                                        |
| database_name   | String | 数据库名称。获取方法请参见获取数据源中的表。                                                                                |
| table_id        | String | 数据表id，获取方法请参见获取数据源中的表。                                                                                |
| table_name      | String | 数据表名称, 获取方法请参见获取数据源中的表。                                                                               |
| user_groups     | String | 用户组列表，用户组名称逗号分隔（非必填项，但用户、用户组必须二选其一进行配置）。例如："userGroup1,userGroup2"。                                   |
| users           | String | 用户列表，用户名称逗号分隔（非必填项，但用户、用户组必须二选其一进行配置），例如："user1,user2"。                                               |

| 参数          | 参数类型                                                  | 描述                                                                                                                                                                                                                                                                                                                        |
|-------------|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| conn_name   | String                                                | 数据连接名称，获取方法请参见 <a href="#">查询数据连接列表</a> 。                                                                                                                                                                                                                                                                                 |
| conn_id     | String                                                | 数据连接id，获取方法请参见 <a href="#">查询数据连接列表</a> 。                                                                                                                                                                                                                                                                                 |
| sync_status | String                                                | 同步状态： <ul style="list-style-type: none"> <li>• UNKNOWN 未知状态</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> <li>• SYNC_PARTIAL_FAIL 存在失败</li> <li>• DELETE_FAIL 删除失败</li> <li>• DELETING 删除中</li> <li>• UPDATING 更新中</li> <li>• DATA_UPDATED 数据存在更新</li> </ul> |
| sync_msg    | String                                                | 策略同步信息。                                                                                                                                                                                                                                                                                                                   |
| sync_log    | String                                                | 同步运行日志, 格式为 字段同步信息+换行符。                                                                                                                                                                                                                                                                                                   |
| create_time | Long                                                  | 策略创建时间。                                                                                                                                                                                                                                                                                                                   |
| create_user | String                                                | 策略创建者。                                                                                                                                                                                                                                                                                                                    |
| update_time | Long                                                  | 策略更新时间。                                                                                                                                                                                                                                                                                                                   |
| update_user | String                                                | 策略更新者。                                                                                                                                                                                                                                                                                                                    |
| schema_name | String                                                | DWS数据源的模式名称。                                                                                                                                                                                                                                                                                                              |
| policy_list | Array of <a href="#">DynamicMaskingPolicy</a> objects | 动态数据脱敏策略列表。                                                                                                                                                                                                                                                                                                               |

表 12-366 DynamicMaskingPolicy

| 参数            | 参数类型   | 描述           |
|---------------|--------|--------------|
| id            | String | 字段脱敏策略id。    |
| policy_set_id | String | 动态脱敏策略id。    |
| column_name   | String | 数据表中的字段名称。   |
| column_type   | String | 数据表中字段的数据类型。 |

| 参数               | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| algorithm_type   | String | <p>动态脱敏规则参数介绍，具体请参考用户指南手册中的“动态脱敏规则介绍”章节。</p> <p>HIVE数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> <p>DWS数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• DWS_ALL_MASK 全掩码</li> <li>• DWS_BACK_KEEP 保留后4位，其余脱敏为*</li> <li>• DWS_FRONT_KEEP 保留前2位，其余脱敏为*</li> <li>• DWS_SELF_CONFIG 需要输入开始位置、结束位置、脱敏字符传入detail结构体中，例如 {"start": 1, "end": 2, "string_target": "*"}</li> </ul> <p>DLI数据源动态脱敏算法</p> <ul style="list-style-type: none"> <li>• MASK 掩盖英文字符和数字</li> <li>• MASK_SHOW_LAST_4 保留后四位</li> <li>• MASK_SHOW_FIRST_4 保留前四位</li> <li>• MASK_HASH 哈希掩盖</li> <li>• MASK_DATE_SHOW_YEAR 掩盖月份和日期</li> <li>• MASK_NULL NULL掩盖</li> </ul> |
| sync_status      | String | <p>同步状态：</p> <ul style="list-style-type: none"> <li>• UNKNOWN 未知状态</li> <li>• NOT_SYNC 未同步</li> <li>• SYNCING 同步中</li> <li>• SYNC_SUCCESS 同步成功</li> <li>• SYNC_FAIL 同步失败</li> <li>• SYNC_PARTIAL_FAIL 存在失败</li> <li>• DELETE_FAIL 删除失败</li> <li>• DELETING 删除中</li> <li>• UPDATING 更新中</li> <li>• DATA_UPDATED 数据存在更新</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| algorithm_detail | String | 动态脱敏策略算法详情。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |



| 参数                   | 参数类型                                | 描述         |
|----------------------|-------------------------------------|------------|
| algorithm_detail_dto | <b>AlgorithmDetailDTO</b><br>object | 动态脱敏策略算法详情 |

表 12-367 AlgorithmDetailDTO

| 参数            | 参数类型    | 描述                    |
|---------------|---------|-----------------------|
| start         | Integer | 开始位数，该值需要大于0且小于end值。  |
| end           | Integer | 结束位数，该值需要大于或等于start值。 |
| int_target    | Integer | 数值类型。                 |
| string_target | String  | 字符串类型。可输入参数为"*"或者"#". |

状态码： 400

表 12-368 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

- 更新HIVE数据源动态脱敏策略  
/v1/0833a5737480d53b2f25c010dc1a7b88/security/masking/dynamic/policies/f52f0f7fe2074421e473b88cdb7752ab

```
{
  "name": "OpenAPI_HIVE_test",
  "datasource_type": "HIVE",
  "cluster_id": "dc425074-26b3-479c-9e2f-b103c0cdd90f",
  "cluster_name": "mrs_3x_autotest_do_not_del",
  "database_name": "bigdatatest",
  "table_name": "chuan",
  "table_id": "NativeTable-ae55cb5c44be4119b8958a3ba5d9a71f-bigdatatest-chuan",
  "user_groups": "DataArts User",
  "users": "",
  "conn_name": "hive_xxxx_0520",
  "conn_id": "ae55cb5c44be4119b8958a3ba5d9a71f",
  "policy_list": [ {
    "column_name": "rr",
    "column_type": "string",
    "algorithm_type": "MASK_SHOW_FIRST_4"
  } ]
}
```

- 更新DWS数据源动态脱敏策略  
/v1/0833a5737480d53b2f25c010dc1a7b88/security/masking/dynamic/policies/6c21c00bbabf1941eebbc2d5a586b2fa

```
{
  "name": "OpenAPI_DWS_test",
  "datasource_type": "DWS",
  "cluster_id": "c94a9133-ef65-47ad-9483-bfa8c5eaa101",
  "cluster_name": "dws_ssl_xxxx_nomodify",
  "database_name": "gaussdb",
  "table_name": "all_column",
  "table_id": "NativeTable-e2f6c44100654778aee017f8ad9e3ac9-gaussdb-public-all_column",
  "user_groups": "DataArts_PDP5_NoPrivilleage",
  "users": "autotest_tics",
  "conn_name": "dws_0430",
  "conn_id": "e2f6c44100654778aee017f8ad9e3ac9",
  "policy_list": [ {
    "column_name": "avarchar",
    "column_type": "varchar",
    "algorithm_type": "DWS_FRONT_KEEP",
    "algorithm_detail_dto": null
  } ],
  "schema_name": "public"
}
```

- 更新DLI数据源动态脱敏策略

/v1/0833a5737480d53b2f25c010dc1a7b88/security/masking/dynamic/policies/62355f975ff8a2943e6ac8992d0951e7

```
{
  "name": "OpenAPI_DLI_xxxx",
  "datasource_type": "DLI",
  "cluster_id": "DLI",
  "cluster_name": "DLI",
  "database_name": "bi",
  "table_name": "ab",
  "table_id": "NativeTable-619d52a0e6954aa68844f5f010e06ef8-bi-ab",
  "user_groups": "",
  "users": "",
  "conn_name": "xxxx_dli_link",
  "conn_id": "619d52a0e6954aa68844f5f010e06ef8",
  "policy_list": [ {
    "column_name": "name1",
    "column_type": "string",
    "algorithm_type": "MASK_HASH"
  } ]
}
```

## 响应示例

状态码: 200

OK

```
{
  "name": "OpenAPI_HIVE_test",
  "datasource_type": "HIVE",
  "cluster_id": "dc425074-26b3-479c-9e2f-b103c0cdd90f",
  "cluster_name": "mrs_3x_xxxx_do_not_del",
  "database_name": "bigdatatest",
  "table_name": "chuan",
  "table_id": "NativeTable-ae55cb5c44be4119b8958a3ba5d9a71f-bigdatatest-chuan",
  "user_groups": "DataArts User",
  "users": "",
  "conn_name": "hive_xxxx_0520",
  "conn_id": "ae55cb5c44be4119b8958a3ba5d9a71f",
  "policy_list": [ {
    "column_name": "rr",
    "column_type": "string",
    "algorithm_type": "MASK_NULL"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 更新HIVE数据源动态脱敏策略

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateSecurityDynamicMaskingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSecurityDynamicMaskingPolicyRequest request = new
        UpdateSecurityDynamicMaskingPolicyRequest();
        request.withId("{id}");
        DynamicMaskingPolicyUpdateDTO body = new DynamicMaskingPolicyUpdateDTO();
        List<DynamicMaskingPolicyUpdate> listbodyPolicyList = new ArrayList<>();
        listbodyPolicyList.add(
            new DynamicMaskingPolicyUpdate()
                .withColumnName("rr")
                .withColumnType("string")
                .withAlgorithmType("MASK_SHOW_FIRST_4")
        );
        body.withPolicyList(listbodyPolicyList);
        body.withConnId("ae55cb5c44be4119b8958a3ba5d9a71f");
        body.withConnName("hive_xxxx_0520");
        body.withUsers("");
        body.withUserGroups("DataArts User");
        body.withTableName("chuan");
        body.withTableId("NativeTable-ae55cb5c44be4119b8958a3ba5d9a71f-bigdatatest-chuan");
        body.withDatabaseName("bigdatatest");
        body.withClusterName("mrs_3x_autotest_do_not_del");
        body.withClusterId("dc425074-26b3-479c-9e2f-b103c0cdd90f");

        body.withDatasourceType(DynamicMaskingPolicyUpdateDTO.DatasourceTypeEnum.fromValue("HIVE")
        );
        body.withName("OpenAPI_HIVE_test");
```

```

request.withBody(body);
try {
    UpdateSecurityDynamicMaskingPolicyResponse response =
client.updateSecurityDynamicMaskingPolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

- **更新DWS数据源动态脱敏策略**

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateSecurityDynamicMaskingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSecurityDynamicMaskingPolicyRequest request = new
UpdateSecurityDynamicMaskingPolicyRequest();
        request.withId("{id}");
        DynamicMaskingPolicyUpdateDTO body = new DynamicMaskingPolicyUpdateDTO();
        List<DynamicMaskingPolicyUpdate> listbodyPolicyList = new ArrayList<>();
        listbodyPolicyList.add(
            new DynamicMaskingPolicyUpdate()
                .withColumnName("avarchar")
                .withColumnType("varchar")
                .withAlgorithmType("DWS_FRONT_KEEP")
        );
        body.withPolicyList(listbodyPolicyList);
        body.withSchemaName("public");
        body.withConnId("e2f6c44100654778aee017f8ad9e3ac9");
    }
}

```

```

        body.withConnName("dws_0430");
        body.withUsers("autotest_tics");
        body.withUserGroups("DataArts_PDP5_NoPrivilege");
        body.withTableName("all_column");
        body.withTableId("NativeTable-e2f6c44100654778aee017f8ad9e3ac9-gaussdb-public-
all_column");
        body.withDatabaseName("gaussdb");
        body.withClusterName("dws_ssl_xxxx_nomodify");
        body.withClusterId("c94a9133-ef65-47ad-9483-bfa8c5ea101");

body.withDatasourceType(DynamicMaskingPolicyUpdateDTO.DatasourceTypeEnum.fromValue("DWS")
);
    body.withName("OpenAPI_DWS_test");
    request.withBody(body);
    try {
        UpdateSecurityDynamicMaskingPolicyResponse response =
client.updateSecurityDynamicMaskingPolicy(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
}

```

- **更新DLI数据源动态脱敏策略**

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateSecurityDynamicMaskingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSecurityDynamicMaskingPolicyRequest request = new

```

```
UpdateSecurityDynamicMaskingPolicyRequest();
    request.withId("{id}");
    DynamicMaskingPolicyUpdateDTO body = new DynamicMaskingPolicyUpdateDTO();
    List<DynamicMaskingPolicyUpdate> listbodyPolicyList = new ArrayList<>();
    listbodyPolicyList.add(
        new DynamicMaskingPolicyUpdate()
            .withColumnName("name1")
            .withColumnType("string")
            .withAlgorithmType("MASK_HASH")
    );
    body.withPolicyList(listbodyPolicyList);
    body.withConnId("619d52a0e6954aa68844f5f010e06ef8");
    body.withConnName("xxx_dli_link");
    body.withUsers("");
    body.withUserGroups("");
    body.withTableName("ab");
    body.withTableId("NativeTable-619d52a0e6954aa68844f5f010e06ef8-bi-ab");
    body.withDatabaseName("bi");
    body.withClusterName("DLI");
    body.withClusterId("DLI");

body.withDatasourceType(DynamicMaskingPolicyUpdateDTO.DatasourceTypeEnum.fromValue("DLI"));
body.withName("OpenAPI_DLI_xxx");
request.withBody(body);
try {
    UpdateSecurityDynamicMaskingPolicyResponse response =
client.updateSecurityDynamicMaskingPolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}
```

## Python

- 更新HIVE数据源动态脱敏策略

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = UpdateSecurityDynamicMaskingPolicyRequest()
    request.id = "{id}"
    listPolicyListbody = [
        DynamicMaskingPolicyUpdate(
            column_name="rr",
            column_type="string",
            algorithm_type="MASK_SHOW_FIRST_4"
        )
    ]
    request.body = DynamicMaskingPolicyUpdateDTO(
        policy_list=listPolicyListbody,
        conn_id="ae55cb5c44be4119b8958a3ba5d9a71f",
        conn_name="hive_xxxx_0520",
        users="",
        user_groups="DataArts User",
        table_name="chuan",
        table_id="NativeTable-ae55cb5c44be4119b8958a3ba5d9a71f-bigdatatest-chuan",
        database_name="bigdatatest",
        cluster_name="mrs_3x_autotest_do_not_del",
        cluster_id="dc425074-26b3-479c-9e2f-b103c0cdd90f",
        datasource_type="HIVE",
        name="OpenAPI_HIVE_test"
    )
    response = client.update_security_dynamic_masking_policy(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 更新DWS数据源动态脱敏策略

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecurityDynamicMaskingPolicyRequest()
        request.id = "{id}"
        listPolicyListbody = [
            DynamicMaskingPolicyUpdate(
                column_name="avarchar",
                column_type="varchar",
                algorithm_type="DWS_FRONT_KEEP"
            )
        ]
```

```
]
request.body = DynamicMaskingPolicyUpdateDTO(
    policy_list=listPolicyListbody,
    schema_name="public",
    conn_id="e2f6c44100654778aee017f8ad9e3ac9",
    conn_name="dws_0430",
    users="autotest_tics",
    user_groups="DataArts_PDP5_NoPrivilege",
    table_name="all_column",
    table_id="NativeTable-e2f6c44100654778aee017f8ad9e3ac9-gaussdb-public-all_column",
    database_name="gaussdb",
    cluster_name="dws_ssl_xxxx_nomody",
    cluster_id="c94a9133-ef65-47ad-9483-bfa8c5eaa101",
    datasource_type="DWS",
    name="OpenAPI_DWS_test"
)
response = client.update_security_dynamic_masking_policy(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 更新DLI数据源动态脱敏策略

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecurityDynamicMaskingPolicyRequest()
        request.id = "{id}"
        listPolicyListbody = [
            DynamicMaskingPolicyUpdate(
                column_name="name1",
                column_type="string",
                algorithm_type="MASK_HASH"
            )
        ]
    ]
    request.body = DynamicMaskingPolicyUpdateDTO(
        policy_list=listPolicyListbody,
        conn_id="619d52a0e6954aa68844f5f010e06ef8",
        conn_name="xxxx_dli_link",
        users="",
        user_groups="",
        table_name="ab",
        table_id="NativeTable-619d52a0e6954aa68844f5f010e06ef8-bi-ab",
        database_name="bi",
```



```
        cluster_name="DLI",
        cluster_id="DLI",
        datasource_type="DLI",
        name="OpenAPI_DLI_xxxx"
    )
    response = client.update_security_dynamic_masking_policy(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

- 更新HIVE数据源动态脱敏策略

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecurityDynamicMaskingPolicyRequest{
        request.Id = "{id}"
        algorithmTypePolicyList:= "MASK_SHOW_FIRST_4"
        var listPolicyListbody = []model.DynamicMaskingPolicyUpdate{
            {
                ColumnName: "rr",
                ColumnType: "string",
                AlgorithmType: &algorithmTypePolicyList,
            },
        },
    }
    usersDynamicMaskingPolicyUpdateDto:= ""
    userGroupsDynamicMaskingPolicyUpdateDto:= "DataArts User"
    tableIdDynamicMaskingPolicyUpdateDto:= "NativeTable-ae55cb5c44be4119b8958a3ba5d9a71f-
bigdatatest-chuan"
    request.Body = &model.DynamicMaskingPolicyUpdateDto{
        PolicyList: listPolicyListbody,
        ConnId: "ae55cb5c44be4119b8958a3ba5d9a71f",
        ConnName: "hive_xxxx_0520",
        Users: &usersDynamicMaskingPolicyUpdateDto,
        UserGroups: &userGroupsDynamicMaskingPolicyUpdateDto,
```

```
    TableName: "chuan",
    TableId: &tableIdDynamicMaskingPolicyUpdateDto,
    DatabaseName: "bigdatatest",
    ClusterName: "mrs_3x_autotest_do_not_del",
    ClusterId: "dc425074-26b3-479c-9e2f-b103c0cdd90f",
    DatasourceType: model.GetDynamicMaskingPolicyUpdateDtoDatasourceTypeEnum().HIVE,
    Name: "OpenAPI_HIVE_test",
  }
  response, err := client.UpdateSecurityDynamicMaskingPolicy(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}
```

- 更新DWS数据源动态脱敏策略

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecurityDynamicMaskingPolicyRequest{}
    request.Id = "{id}"
    algorithmTypePolicyList := "DWS_FRONT_KEEP"
    var listPolicyListbody = []model.DynamicMaskingPolicyUpdate{
        {
            ColumnName: "avarchar",
            ColumnType: "varchar",
            AlgorithmType: &algorithmTypePolicyList,
        },
    }
    schemaNameDynamicMaskingPolicyUpdateDto := "public"
    usersDynamicMaskingPolicyUpdateDto := "autotest_tics"
    userGroupsDynamicMaskingPolicyUpdateDto := "DataArts_PDP5_NoPrivilege"
    tableIdDynamicMaskingPolicyUpdateDto := "NativeTable-e2f6c44100654778aee017f8ad9e3ac9-
    gaussdb-public-all_column"
    request.Body = &model.DynamicMaskingPolicyUpdateDto{
        PolicyList: listPolicyListbody,
        SchemaName: &schemaNameDynamicMaskingPolicyUpdateDto,
        ConnId: "e2f6c44100654778aee017f8ad9e3ac9",
        ConnName: "dws_0430",
    }
}
```

```

Users: &usersDynamicMaskingPolicyUpdateDto,
UserGroups: &userGroupsDynamicMaskingPolicyUpdateDto,
TableName: "all_column",
TableId: &tableIdDynamicMaskingPolicyUpdateDto,
DatabaseName: "gaussdb",
ClusterName: "dws_ssl_xxxx_nomodify",
ClusterId: "c94a9133-ef65-47ad-9483-bfa8c5eaa101",
DatasourceType: model.GetDynamicMaskingPolicyUpdateDtoDatasourceTypeEnum().DWS,
Name: "OpenAPI_DWS_test",
}
response, err := client.UpdateSecurityDynamicMaskingPolicy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

- 更新DLI数据源动态脱敏策略

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecurityDynamicMaskingPolicyRequest{
        request.Id = "{id}"
        algorithmTypePolicyList:= "MASK_HASH"
        var listPolicyListbody = []model.DynamicMaskingPolicyUpdate{
            {
                ColumnName: "name1",
                ColumnType: "string",
                AlgorithmType: &algorithmTypePolicyList,
            },
        },
    }
    usersDynamicMaskingPolicyUpdateDto:= ""
    userGroupsDynamicMaskingPolicyUpdateDto:= ""
    tableIdDynamicMaskingPolicyUpdateDto:= "NativeTable-619d52a0e6954aa68844f5f010e06ef8-bi-ab"
    request.Body = &model.DynamicMaskingPolicyUpdateDto{
        PolicyList: listPolicyListbody,
        ConnId: "619d52a0e6954aa68844f5f010e06ef8",
        ConnName: "xxxx_dli_link",
    }
}

```

```

Users: &usersDynamicMaskingPolicyUpdateDto,
UserGroups: &userGroupsDynamicMaskingPolicyUpdateDto,
TableName: "ab",
TableId: &tableIdDynamicMaskingPolicyUpdateDto,
DatabaseName: "bi",
ClusterName: "DLI",
ClusterId: "DLI",
DatasourceType: model.GetDynamicMaskingPolicyUpdateDtoDatasourceTypeEnum().DLI,
Name: "OpenAPI_DLI_xxxx",
}
response, err := client.UpdateSecurityDynamicMaskingPolicy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

## 12.14.5 批量删除动态脱敏策略

### 功能介绍

批量删除动态脱敏策略。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/masking/dynamic/policies/batch-delete

表 12-369 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-370 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-371 请求 Body 参数

| 参数  | 是否必选 | 参数类型             | 描述             |
|-----|------|------------------|----------------|
| ids | 否    | Array of strings | 待删除动态脱敏策略id列表。 |

## 响应参数

状态码： 400

表 12-372 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/masking/dynamic/policies/batch-delete
{
  "ids": [ "7ea6b287dfd34b81c5eb627de2fd3401" ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dataartsstudio.v1.region.DataArtsStudioRegion;
import com.huaweicloud.sdk.dataartsstudio.v1.*;
import com.huaweicloud.sdk.dataartsstudio.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteSecurityDynamicMaskingPoliciesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DataArtsStudioClient client = DataArtsStudioClient.newBuilder()
            .withCredential(auth)
            .withRegion(DataArtsStudioRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchDeleteSecurityDynamicMaskingPoliciesRequest request = new
        BatchDeleteSecurityDynamicMaskingPoliciesRequest();
        BatchDeletePolicySetsDTO body = new BatchDeletePolicySetsDTO();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("7ea6b287dfd34b81c5eb627de2fd3401");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            BatchDeleteSecurityDynamicMaskingPoliciesResponse response =
            client.batchDeleteSecurityDynamicMaskingPolicies(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdataartsstudio.v1.region.dataartsstudio_region import DataArtsStudioRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdataartsstudio.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DataArtsStudioClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DataArtsStudioRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteSecurityDynamicMaskingPoliciesRequest()
        listIdsbody = [
            "7ea6b287dfd34b81c5eb627de2fd3401"
        ]
        request.body = BatchDeletePolicySetsDTO(
            ids=listIdsbody
        )
        response = client.batch_delete_security_dynamic_masking_policies(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dataartsstudio "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dataartsstudio/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dataartsstudio.NewDataArtsStudioClient(
        dataartsstudio.DataArtsStudioClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```

request := &model.BatchDeleteSecurityDynamicMaskingPoliciesRequest{}
var listIdsbody = []string{
    "7ea6b287dfd34b81c5eb627de2fd3401",
}
request.Body = &model.BatchDeletePolicySetsDto{
    Ids: &listIdsbody,
}
response, err := client.BatchDeleteSecurityDynamicMaskingPolicies(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | No Content  |
| 400 | Bad Request |

## 12.15 权限审批接口

### 12.15.1 获取表权限审批人列表

#### 功能介绍

获取表权限审批人列表

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/{project\_id}/security/permission-application/table/approvers

表 12-373 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |



## 请求参数

表 12-374 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-375 请求 Body 参数

| 参数              | 是否必选 | 参数类型   | 描述                                |
|-----------------|------|--------|-----------------------------------|
| instance_id     | 否    | String | 实例id                              |
| datasource_type | 否    | String | 数据源类型,hive,dws,dli                |
| cluster_id      | 否    | String | 集群id,dli传DLI，dws和mrs-hive传对应的集群id |
| schema_name     | 否    | String | schema名称，dws需要传这个字段               |
| database_name   | 否    | String | 数据库名称                             |
| table_name      | 否    | String | 表名称                               |
| expire_time     | 否    | Long   | 权限到期时间戳，毫秒。                       |

## 响应参数

状态码： 200

表 12-376 响应 Body 参数

| 参数             | 参数类型                                           | 描述      |
|----------------|------------------------------------------------|---------|
| approvers      | Array of <a href="#">TableApprover</a> objects | 审批人列表   |
| has_table_rule | Boolean                                        | 是否已经有权限 |

表 12-377 TableApprover

| 参数                  | 参数类型    | 描述                                                                                                  |
|---------------------|---------|-----------------------------------------------------------------------------------------------------|
| id                  | String  | 审批人id                                                                                               |
| name                | String  | 审批人姓名                                                                                               |
| permission_set_id   | String  | 权限集id                                                                                               |
| type                | Integer | 审批人类型, 0 用户 1 用户组 2 角色 (空间管理员)                                                                      |
| workspace_id        | String  | 空间id                                                                                                |
| table_approver_type | String  | 安全管理员: SECURITY_MANAGER, 空间权限集管理员<br>WORKSPACE_PERMISSION_SET_MANAGER, 权限集管理员PERMISSION_SET_MANAGER |

状态码: 400

表 12-378 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

### 请求示例

无

### 响应示例

无

### 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.15.2 提交表权限申请

### 功能介绍

提交表权限申请

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-application/table

表 12-379 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

### 请求参数

表 12-380 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-381 请求 Body 参数

| 参数         | 是否必选 | 参数类型                                                 | 描述                                                                         |
|------------|------|------------------------------------------------------|----------------------------------------------------------------------------|
| approver   | 否    | <a href="#">TableApprover</a> object                 | 申请审批人，由 /v1/{project_id}/security/permission-application/table/approvers返回 |
| table_info | 否    | <a href="#">ListTableApproversRequestBody</a> object | 申请的表信息                                                                     |
| proposers  | 否    | Array of <a href="#">TableProposers</a> objects      | 申请人列表                                                                      |

| 参数     | 是否必选 | 参数类型   | 描述   |
|--------|------|--------|------|
| reason | 否    | String | 申请理由 |

表 12-382 TableApprover

| 参数                  | 是否必选 | 参数类型    | 描述                                                                                                         |
|---------------------|------|---------|------------------------------------------------------------------------------------------------------------|
| id                  | 否    | String  | 审批人id                                                                                                      |
| name                | 否    | String  | 审批人姓名                                                                                                      |
| permission_set_id   | 否    | String  | 权限集id                                                                                                      |
| type                | 否    | Integer | 审批人类型, 0 用户 1 用户组 2 角色 (空间管理员)                                                                             |
| workspace_id        | 否    | String  | 空间id                                                                                                       |
| table_approver_type | 否    | String  | 安全管理员:<br>SECURITY_MANAGER, 空间权限集管理员<br>WORKSPACE_PERMISSION_SET_MANAGER, 权限集管理员<br>PERMISSION_SET_MANAGER |

表 12-383 ListTableApproversRequestBody

| 参数              | 是否必选 | 参数类型   | 描述                                 |
|-----------------|------|--------|------------------------------------|
| instance_id     | 否    | String | 实例id                               |
| datasource_type | 否    | String | 数据源类型,hive,dws,dli                 |
| cluster_id      | 否    | String | 集群id,dli传DLI, dws和mrs-hive传对应的集群id |
| schema_name     | 否    | String | schema名称, dws需要传这个字段               |
| database_name   | 否    | String | 数据库名称                              |
| table_name      | 否    | String | 表名称                                |
| expire_time     | 否    | Long   | 权限到期时间戳, 毫秒。                       |

表 12-384 TableProposers

| 参数   | 是否必选 | 参数类型    | 描述                |
|------|------|---------|-------------------|
| id   | 否    | String  | 申请人id, 权限组id      |
| name | 否    | String  | 申请人姓名             |
| type | 否    | Integer | 审批人类型, 0 用户 1 用户组 |

## 响应参数

状态码： 200

表 12-385 响应 Body 参数

| 参数                    | 参数类型   | 描述     |
|-----------------------|--------|--------|
| describe              | String | 描述     |
| permission_center_url | String | 审批页面地址 |
| workspace_id          | String | 工作空间id |

状态码： 400

表 12-386 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

无

## 响应示例

无

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | ok |

| 状态码 | 描述          |
|-----|-------------|
| 400 | Bad Request |

## 12.15.3 审批通过工单

### 功能介绍

审批通过工单。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-approve/approve/{id}

表 12-387 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 工单id                                     |

### 请求参数

表 12-388 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-389 请求 Body 参数

| 参数     | 是否必选 | 参数类型   | 描述        |
|--------|------|--------|-----------|
| reason | 是    | String | 审批通过/驳回原因 |

## 响应参数

状态码： 400

表 12-390 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

审批通过工单id1

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-approve/approve/id1  
{  
  "reason": "同意"  
}
```

## 响应示例

无

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | noContent   |
| 400 | Bad Request |

## 12.15.4 驳回工单

### 功能介绍

驳回工单。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/security/permission-approve/reject/{id}

表 12-391 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |
| id         | 是    | String | 工单id                                     |

## 请求参数

表 12-392 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

表 12-393 请求 Body 参数

| 参数     | 是否必选 | 参数类型   | 描述        |
|--------|------|--------|-----------|
| reason | 是    | String | 审批通过/驳回原因 |

## 响应参数

状态码： 400

表 12-394 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

驳回工单id1

```
/v1/0833a5737480d53b2f25c010dc1a7b88/security/permission-approve/reject/id1
{
```



```
"reason": "驳回"  
}
```

## 响应示例

无

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | noContent   |
| 400 | Bad Request |

## 12.15.5 获取审批工单列表

### 功能介绍

获取审批工单列表

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/security/openapi/permission-approve/approvals

表 12-395 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

表 12-396 Query 参数

| 参数            | 是否必选 | 参数类型    | 描述                       |
|---------------|------|---------|--------------------------|
| limit         | 否    | Integer | 每页显示的条目数量。               |
| offset        | 否    | Integer | 偏移量，表示从此偏移量开始查询，该值大于等于0。 |
| proposer_name | 否    | String  | 申请人名称                    |
| approval_id   | 否    | String  | 工单id                     |

| 参数                     | 是否必选 | 参数类型              | 描述                        |
|------------------------|------|-------------------|---------------------------|
| workspace_id           | 否    | String            | 工作空间id                    |
| status_list            | 否    | Array of integers | 工单状态                      |
| application_start_time | 否    | Long              | 申请开始时间                    |
| application_end_time   | 否    | Long              | 申请结束时间                    |
| order_by_desc          | 否    | Boolean           | 升降序                       |
| order_by               | 否    | String            | 排序参数, START_TIME,END_TIME |

## 请求参数

表 12-397 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                      |
|--------------|------|--------|-------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID, 获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token, 通过调用IAM服务获取用户Token接口获取(响应消息头中X-Subject-Token的值)使用Token认证时必选。 |

## 响应参数

状态码: 200

表 12-398 响应 Body 参数

| 参数        | 参数类型                                                          | 描述    |
|-----------|---------------------------------------------------------------|-------|
| approvals | Array of <a href="#">PermissionApprovalOpenapiDTO</a> objects | 工单列表  |
| total     | Long                                                          | 规则组总数 |

表 12-399 PermissionApprovalOpenapiDTO

| 参数                              | 参数类型                                                | 描述                                              |
|---------------------------------|-----------------------------------------------------|-------------------------------------------------|
| approval_dispatch_error_message | String                                              | 审批外发失败消息                                        |
| approval_dispatch_status        | String                                              | 审批外发状态, 0表示成功, 1表示失败, null表示非SMN节点              |
| approval_type                   | String                                              | 申请类型, DATA_PERMISSION                           |
| approve_reason                  | String                                              | 申请原因                                            |
| current_node_id                 | String                                              | 当前审批节点id                                        |
| current_node_name               | String                                              | 当前审批节点审批人                                       |
| current_node_type               | String                                              | 当前审批节点审批人类型                                     |
| detail                          | <a href="#">PermissionApprovalDetail</a> DTO object | 审批详细信息                                          |
| end_time                        | String                                              | 工单结束时间                                          |
| expire_time                     | String                                              | 到期时间                                            |
| id                              | String                                              | 工单id                                            |
| instance_id                     | String                                              | 实例id                                            |
| permission_set_id               | String                                              | 审批人所在权限集id                                      |
| project_id                      | String                                              | 项目id                                            |
| proposer_id                     | String                                              | 申请人id                                           |
| proposer_name                   | String                                              | 申请人名称                                           |
| proposer_workspace_id           | String                                              | 用户申请权限时所在工作空间id                                 |
| reason                          | String                                              | 拒绝理由                                            |
| start_time                      | Integer                                             | 工单开始时间                                          |
| status                          | String                                              | 工单状态, WAITING_APPROVE, APPROVED, REJECT, REVOKE |
| workspace_id                    | String                                              | 工作空间id                                          |

| 参数             | 参数类型   | 描述     |
|----------------|--------|--------|
| workspace_name | String | 工作空间名称 |

表 12-400 PermissionApprovalDetailDTO

| 参数              | 参数类型                                | 描述                                                                                                    |
|-----------------|-------------------------------------|-------------------------------------------------------------------------------------------------------|
| cluster_id      | String                              | 集群id                                                                                                  |
| cluster_name    | String                              | 集群名称                                                                                                  |
| datasource_type | String                              | 数据源类型 <ul style="list-style-type: none"> <li>• HIVE数据源</li> <li>• DWS数据源</li> <li>• DLI数据源</li> </ul> |
| expire_time     | String                              | 超时时间                                                                                                  |
| permissions     | Array of <b>permissions</b> objects | 申请权限详情列表                                                                                              |
| proposers       | Array of <b>proposers</b> objects   | 申请人详情列表                                                                                               |

表 12-401 permissions

| 参数                | 参数类型             | 描述           |
|-------------------|------------------|--------------|
| column_name       | String           | 列名称          |
| database_name     | String           | 库名称          |
| permission_action | Array of strings | 权限           |
| permission_set_id | String           | 申请权限所在的空间权限集 |
| schema_name       | String           | schema名称     |
| secrecy_level_id  | String           | 密级           |
| table_name        | String           | 表名           |

表 12-402 proposers

| 参数   | 参数类型   | 描述                                                   |
|------|--------|------------------------------------------------------|
| id   | String | 申请人id                                                |
| name | String | 申请人名称                                                |
| type | String | 申请人类<br>型,USER,USER_GROUP,ROLE,WORKSPACE_ACC<br>OUNT |

状态码： 400

表 12-403 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

/v1/{project\_id}/security/openapi/permission-approve/approvals

## 响应示例

无

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

## 12.16 权限申请接口

### 12.16.1 撤回工单申请

#### 功能介绍

撤回工单申请。

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1/{project\_id}/security/permission-application/applications/{id}

表 12-404 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| id         | 是    | String | 申请工单id                                   |
| project_id | 是    | String | 项目ID，获取方法请参见 <a href="#">项目ID和账号ID</a> 。 |

## 请求参数

表 12-405 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                     |
|--------------|------|--------|------------------------------------------------------------------------|
| workspace    | 是    | String | 工作空间ID，获取方法请参见 <a href="#">实例ID和工作空间ID</a> 。                           |
| X-Auth-Token | 是    | String | IAM Token，通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）使用Token认证时必选。 |

## 响应参数

状态码： 400

表 12-406 响应 Body 参数

| 参数         | 参数类型   | 描述    |
|------------|--------|-------|
| error_code | String | 错误码。  |
| error_msg  | String | 错误描述。 |

## 请求示例

无

## 响应示例

无

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | ok          |
| 400 | Bad Request |

# 13 应用示例

## 13.1 数据集成应用示例

本节通过cURL调用CDM API，迁移本地MySQL数据库中的数据到云上服务DWS为例，介绍使用CDM API的基本流程。

### 1. 获取token

获取用户的token，因为在后续的请求中需要将token放到请求消息头中作为认证。

### 2. 创建CDM集群

- 如果您已经创建过CDM集群，可以跳过该步骤，直接使用已创建的集群ID。
- 如果您需要使用新的集群执行迁移任务，调用[创建集群](#)API创建。

### 3. 创建连接

调用[创建连接](#)API创建MySQL连接和DWS连接。

### 4. 创建迁移作业

调用[指定集群创建作业](#)API创建MySQL到DWS的迁移作业。

### 5. 查看作业结果

调用[启动作业](#)API开始执行作业。

## 准备数据

在调用API之前，您需要准备如下数据。

表 13-1 准备数据

| 数据项   | 名称   | 说明           | 样例                               |
|-------|------|--------------|----------------------------------|
| 云账户信息 | 项目名  | CDM所属的项目名。   | Project Name                     |
|       | 项目ID | CDM所属的项目ID。  | 1551c7f6c808414d8e9f3c514a170f2e |
|       | 账号名  | 用户所属的企业账户名称。 | Account Name                     |



| 数据项      | 名称           | 说明                                                                                                   | 样例                                   |
|----------|--------------|------------------------------------------------------------------------------------------------------|--------------------------------------|
|          | 用户名          | 使用云服务的用户名，该用户需要拥有CDM的操作权限。                                                                           | Username                             |
|          | 密码           | 用户密码。                                                                                                | password                             |
| VPC信息    | VPC的ID       | CDM所属的VPC必须与DWS一致。                                                                                   | 6b47302a-bf79-4b20-bf7a-80987408e196 |
|          | 子网ID         | CDM所属的子网必须与DWS一致。                                                                                    | 63bdc3cb-a4e7-486f-82ee-d9bf208c8f8c |
|          | 安全组ID        | CDM所属的安全组必须与DWS一致。                                                                                   | 005af77a-cce5-45ac-99c7-2ea50ea8addf |
| Endpoint | IAM的Endpoint | 终端节点（Endpoint）即调用API的 <b>请求地址</b> ，不同服务不同区域的终端节点不同。Endpoint您可以从 <a href="#">终端节点及区域说明</a> 获取。        | iam_endpoint                         |
|          | CDM的Endpoint | 终端节点（Endpoint）即调用API的 <b>请求地址</b> ，不同服务不同区域的终端节点不同。本服务的Endpoint您可以从 <a href="#">终端节点Endpoint</a> 获取。 | cdm_endpoint                         |
| MySQL数据库 | IP地址         | 本地的MySQL数据库的IP地址，且该地址允许CDM通过公网IP访问。                                                                  | 1xx.120.85.24                        |
|          | 端口           | MySQL数据库的端口。                                                                                         | 3306                                 |
|          | 数据库名称        | 待导出数据的MySQL数据库名称。                                                                                    | DB_name                              |
|          | 用户名          | 访问MySQL数据库的用户，该用户拥有MySQL数据库的读、写和删除权限。                                                                | username                             |
|          | 密码           | 访问MySQL数据库的用户密码。                                                                                     | DB_password                          |
| DWS数据库   | IP地址         | DWS数据库的IP地址，CDM可通过内网访问该地址。                                                                           | 10.120.85.24                         |
|          | 端口           | DWS数据库的端口。                                                                                           | 3306                                 |

| 数据项 | 名称    | 说明                                | 样例           |
|-----|-------|-----------------------------------|--------------|
|     | 数据库名称 | 待写入数据的DWS数据库名称。                   | DWS          |
|     | 用户名   | 访问DWS数据库的用户，该用户拥有DWS数据库的读、写和删除权限。 | user_dws     |
|     | 密码    | 访问DWS数据库的用户密码。                    | dws_password |

## 获取 token

1. 调用其他API前，需要获取token，并设置成环境变量。

```
curl -H "Content-Type:application/json" https://{iam_endpoint}/v3/auth/tokens -X POST -d '{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "Username",
          "password": "password",
          "domain": {
            "name": "Account Name"
          }
        }
      }
    },
    "scope": {
      "project": {
        "id": "1551c7f6c808414d8e9f3c514a170f2e"
      }
    }
  }
}' -v -k
```

响应Header中“X-Subject-Token”的值即为Token：

```
X-Subject-Token:MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...
```

2. 使用如下命令将token设置为环境变量，方便后续事项。

```
export Token = MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...
```

## 创建 CDM 集群

1. 调用[创建集群](#)API创建集群，假设集群详情如下：

- 集群名称为“cdm-ab82”。
- 集群规格为“cdm.medium”。
- VPC、子网、安全组与DWS一致，且自动绑定弹性IP。

如果返回状态码为200，则说明创建命令执行成功。

```
curl -X POST -H 'Content-Type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '{
  "cluster": {
    "name": "cdm-ab82",
```

```

"vpclid": "6b47302a-bf79-4b20-bf7a-80987408e196",
"instances": [{
  "flavorRef": "fb8fe666-6734-4b11-bc6c-43d11db3c745",
  "nics": [{
    "net-id": "63bdc3cb-a4e7-486f-82ee-d9bf208c8f8c",
    "securityGroupId": "005af77a-cce5-45ac-99c7-2ea50ea8addf"
  }],
  "availability_zone": "Project Name",
  "type": "cdm"
}],
"datastore": {
  "version": "1.8.5",
  "type": "cdm"
},
"isScheduleBootOff": false,
"scheduleBootTime": "null",
"scheduleOffTime": "null",
"isAutoOff": false,
"sys_tags": [{
  "key": "_sys_enterprise_project_id",
  "value": "1ce45885-4033-40d2-bdde-d4dbaceb387d"
}]
},
"autoRemind": false,
"phoneNum": "null",
"email": "null"
}
https://{cdm_endpoint}/v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters -v -k

```

2. 调用[查询集群列表](#)查询集群信息，获取集群的ID，并设置为全局变量。

```

curl -X GET -H 'Content-Type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://{cdm_endpoint}/v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters -k -v

```

获取响应如下所示。

```

{
  "clusters": [{
    "version": "x.x.x",
    "updated": "2018-09-05T08:38:25",
    "name": "cdm-ab82",
    "created": "2018-09-05T08:38:25",
    "id": "bae65496-643e-47ca-84af-948672de7eeb",
    "status": "200",
    "isFrozen": "0",
    "statusDetail": "Normal",
    "actionProgress": {},
    "config_status": "In-Sync"
  }]
}

```

“status”的状态如果为200则表示集群创建成功，集群对应的ID为**bae65496-643e-47ca-84af-948672de7eeb**。

3. 使用如下命令将集群对应的ID设置为全局变量，方便后续事项。

```

export ID = bae65496-643e-47ca-84af-948672de7eeb

```

## 创建连接

1. 调用[创建连接](#)API创建MySQL连接，连接名称为*mysql\_link*。这里假设本地MySQL数据库信息如下：

- IP地址为 *1xx.120.85.24*。
- 端口为 *3306*。
- 数据库名称为 *DB\_name*。
- 登录用户为 *username*。
- 密码为 *DB\_password*。

如果返回状态码为200，则说明创建命令执行成功。

```
curl -X POST -H "Content-Type:application/json" -H "X-Auth-Token:$Token" -d '{
  "links": [{
    "enabled": true,
    "update-user": null,
    "name": "mysql_link",
    "link-config-values": {
      "configs": [
        {
          "name": "linkConfig",
          "inputs": [
            {
              "name": "linkConfig.databaseType",
              "value": "MYSQL"
            },
            {
              "name": "linkConfig.host",
              "value": "1xx.120.85.24"
            },
            {
              "name": "linkConfig.port",
              "value": "3306"
            },
            {
              "name": "linkConfig.database",
              "value": "DB_name"
            },
            {
              "name": "linkConfig.username",
              "value": "username"
            },
            {
              "name": "linkConfig.password",
              "value": "DB_password"
            },
            {
              "name": "linkConfig.fetchSize",
              "value": "100000"
            },
            {
              "name": "linkConfig.usingNative",
              "value": "true"
            }
          ]
        }
      ]
    },
    "connector-name": "generic-jdbc-connector",
    "creation-date": 1536654788622,
    "update-date": 1536654788622,
    "creation-user": null
  }
]
}'
https://{cdm_endpoint}/v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/
bae65496-643e-47ca-84af-948672de7eeb/cdm/link -k -v
```

2. 调用[创建连接](#)API创建DWS连接，连接名称为 *dws\_link*。这里假设DWS数据库信息如下：

- 数据库的IP地址为 *10.120.85.24*。
- 端口为 *3306*。
- 数据库的名称为 *DWS*。
- 登录用户为 *user\_dws*。
- 密码为 *dws\_password*。

```
curl -X POST -H "Content-Type:application/json" -H "X-Auth-Token:$Token" -d '{
  "links": [{
    "enabled": true,
    "update-user": null,
```

```

"name": "dws_link",
"link-config-values": {
  "configs": [
    {
      "name": "linkConfig",
      "inputs": [
        {
          "name": "linkConfig.databaseType",
          "value": "DWS"
        },
        {
          "name": "linkConfig.host",
          "value": "10.120.85.24"
        },
        {
          "name": "linkConfig.port",
          "value": "3306"
        },
        {
          "name": "linkConfig.database",
          "value": "DWS"
        },
        {
          "name": "linkConfig.username",
          "value": "user_dws"
        },
        {
          "name": "linkConfig.password",
          "value": "dws_password"
        },
        {
          "name": "linkConfig.fetchSize",
          "value": "100000"
        },
        {
          "name": "linkConfig.usingNative",
          "value": "true"
        }
      ]
    }
  ]
},
"connector-name": "generic-jdbc-connector",
"creation-date": 1536654788622,
"update-date": 1536654788622,
"creation-user": null
}]
}'
https://{cdm_endpoint}/v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/
bae65496-643e-47ca-84af-948672de7eeb/cdm/link -k -v

```

## 创建迁移作业

1. 连接创建成功后，调用[指定集群创建作业](#)API创建迁移作业，作业样例如下：
  - 作业名称：mysql2dws。
  - 从MySQL导出数据的数据库名称为*default*，导出的表名称为*mysql\_tbl*，依据*id*字段将作业分割为多个任务并发执行。
  - 导入DWS的数据库名称为*public*，表名为*cdm\_all\_type*，导入前不清空数据。
  - 当DWS数据库里没有本地MySQL数据库中的表时，CDM自动在DWS端创建该表。
  - DWS端加载的字段列表为*id&gid&name*。
  - 作业抽取数据时，并发执行的Extractor数量为3。

如果返回状态码为200，则说明创建命令执行成功。

```
curl -X POST -H "Content-Type:application/json" -H "X-Cluster-ID:$ID" -H "X-Auth-Token:$Token" -d '{
  "jobs": [{
    "job_type": "NORMAL_JOB",
    "name": "mysql2dws",
    "from-link-name": "mysql_link",
    "from-connector-name": "generic-jdbc-connector",
    "to-link-name": "dws_link",
    "to-connector-name": "generic-jdbc-connector",
    "from-config-values": {
      "configs": [{
        "name": "fromJobConfig",
        "inputs": [{
          "name": "fromJobConfig.schemaName",
          "value": "default"
        }],
        {
          "name": "fromJobConfig.tableName",
          "value": "mysql_tbl"
        },
        {
          "name": "fromJobConfig.partitionColumn",
          "value": "id"
        }
      ]
    }
  },
  "to-config-values": {
    "configs": [
      {
        "inputs": [
          {
            "name": "toJobConfig.schemaName",
            "value": "public"
          },
          {
            "name": "toJobConfig.tablePreparation",
            "value": "CREATE_WHEN_NOT_EXIST"
          },
          {
            "name": "toJobConfig.tableName",
            "value": "cdm_all_type"
          },
          {
            "name": "toJobConfig.columnList",
            "value": "id&gid&name"
          },
          {
            "name": "toJobConfig.shouldClearTable",
            "value": "false"
          }
        ]
      },
      {
        "name": "toJobConfig"
      }
    ]
  },
  "driver-config-values": {
    "configs": [{
      "name": "throttlingConfig",
      "inputs": [{
        "name": "throttlingConfig.numExtractors",
        "value": "3"
      }],
    }],
  }
}]' https://{cdm_endpoint}/v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/
bae65496-643e-47ca-84af-948672de7eeb/cdm/job -k -v
```

2. 调用[启动作业](#)API开始执行作业。

```
curl -X GET -H 'Content-Type:application/json;charset=utf-8' -H "X-Cluster-ID:$ID" -H "X-Auth-Token:$Token" https://{cdm_endpoint}/v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/bae65496-643e-47ca-84af-948672de7eeb/cdm/job/mysql2dws/start -k -v
```

响应如下:

```
{
  "submissions": [{
    "progress": 1,
    "job-name": "mysql2dws",
    "status": "BOOTING",
    "creation-date": 1536654788622,
    "creation-user": "cdm"
  }]
}
```

## 查看作业结果

1. 调用[查询作业状态](#)API查询作业状态。

```
curl -X GET -H 'Content-Type:application/json;charset=utf-8' -H "X-Cluster-ID:$ID" -H "X-Auth-Token:$Token" https://{cdm_endpoint}/v1.1/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/mysql2dws/status -k -v
```

2. 查看作业执行结果, 作业执行成功的响应如下:

```
{
  "submissions": [{
    "progress": 0,
    "job-name": "mysql2dws",
    "status": "SUCCEEDED",
    "creation-date": 1536654788622,
    "creation-user": "cdm",
    "isStoppingIncrement": "",
    "last-update-date": 1536654888622,
    "is-execute-auto": false,
    "last-udpate-user": "cdm",
    "isDeleteJob": false,
    "isIncrementing": false,
    "external-id": "job_local1127970451_0009",
    "counters": {
      "org.apache.sqoop.submission.counter.SqoopCounters": {
        "BYTES_WRITTEN": -1,
        "TOTAL_FILES": -1,
        "BYTES_READ": -1,
        "FILES_WRITTEN": -1,
        "TOTAL_SIZE": -1,
        "FILES_READ": -1,
        "ROWS_WRITTEN": 80,
        "ROWS_READ": 80
      }
    }
  }]
}
```

### 说明

- BYTES\_WRITTEN: 表示写入的字节数。
- BYTES\_READ: 表示读取的字节数。
- TOTAL\_FILES: 表示总文件数。
- FILES\_WRITTEN: 表示写入的文件数。
- FILES\_READ: 表示读取的文件数。
- ROWS\_WRITTEN: 表示写入成功的行数。
- ROWS\_READ: 表示读取成功的行数。

## 13.2 数据开发应用示例

### 操作场景

数据治理中心DataArts Studio帮助企业快速构建从数据接入到数据分析的端到端智能数据系统，消除数据孤岛，统一数据标准，加快数据变现，实现数字化转型。

下面介绍如何调用[创建脚本](#)API创建脚本，API的调用方法请参见[如何调用API](#)。

### 前提条件

您需要规划数据开发服务所在的区域信息，并根据区域确定调用API的Endpoint。

终端节点（Endpoint）即调用API的[请求地址](#)，不同服务不同区域的终端节点不同。本服务的Endpoint您可以从[终端节点Endpoint](#)获取。

### 创建 Shell 脚本

如下示例是创建shell脚本的配置。

```
{
  "name": "echoTimeShell",
  "type": "Shell",
  "content": "echo a",
  "connectionName": "con"
}
```

- name：脚本的名称，由您自行定义，例如取名为echoTimeShell。
- type：脚本类型。
- content：脚本的内容。
- connectionName：脚本关联的连接名称。

### 创建 DLI SQL 脚本

您还可以创建DLI SQL脚本，示例如下。

```
{
  "name": "dliSql1",
  "type": "DLISQL",
  "content": "show tables",
  "connectionName": "dliCon1",
  "database": "testDatabase1",
  "queueName": "queue1"
}
```

- name：脚本的名称，由您自行定义，例如取名为dliSql1。
- type：脚本类型。
- content：脚本的内容。
- connectionName：脚本关联的连接名称。
- database：脚本执行的时候，关联的数据库名称。
- queueName：sql执行的时候，运行在dli的队列名称。



# 14 附录

## 14.1 公共消息头

### 公共请求消息头

公共请求消息头如表14-1所示。

表 14-1 公共请求消息头

| 名称             | 描述                                                                                                                               | 是否必选             | 示例                                                                    |
|----------------|----------------------------------------------------------------------------------------------------------------------------------|------------------|-----------------------------------------------------------------------|
| Content-Type   | 消息体的类型（格式），默认取值为“application/json”。<br>如果请求消息体中含有中文字符，则还需要通过charset=utf8指定中文字符集。                                                 | 是                | application/json;charset=utf8                                         |
| Content-Length | 请求body长度，单位为Byte。                                                                                                                | 否                | 3495                                                                  |
| X-Language     | 请求语言。                                                                                                                            | 否                | en_us                                                                 |
| X-Auth-Token   | 用户Token。用户Token是调用 <b>获取用户Token</b> 接口的响应值，因此调用 <b>获取用户Token</b> 接口是唯一不需要认证的接口。<br>请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 | 使用Token认证时该字段必选。 | 注：以下仅为Token示例片段<br>MIIPAgYJKoZlhvc<br>NAQcCo...ggg1BBI<br>INPXsidG9rZ |

| 名称            | 描述                                                                                          | 是否必选                                                                                                                                                                                                              | 示例                                       |
|---------------|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Authorization | 签名认证信息。<br>当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。                                          | 使用AK/SK认证时必选。                                                                                                                                                                                                     | -                                        |
| X-Sdk-Date    | 请求的发生时间，格式为(YYYYMMDD'T'HHMMSS'Z')，取值为当前系统的GMT时间。<br>当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。 | 使用AK/SK认证时必选。                                                                                                                                                                                                     | 20150907T101459Z                         |
| Host          | 请求的服务器信息，从服务API的URL中获取。值为hostname[:port]。端口缺省时使用默认的端口，https的默认端口为443。                       | 使用AK/SK认证时必选。                                                                                                                                                                                                     | code.test.com<br>or<br>code.test.com:443 |
| X-Project-Id  | 子项目ID。请参考 <a href="#">项目ID和账号ID</a> 章节获取项目ID。                                               | 调用在子项目中的服务接口，并使用AK/SK认证时必选。                                                                                                                                                                                       | e9993fc787d94b6c886cbaa340f9c0f4         |
| X-Domain-ID   | 账号ID。请参考 <a href="#">项目ID和账号ID</a> 章节获取账号ID。                                                | 调用全局服务接口，并使用AK/SK认证时必选。                                                                                                                                                                                           | 0a2f5e28af00f3850f31c00f31e884e0         |
| workspace     | 工作空间id，获取方式请参见 <a href="#">实例ID和工作空间ID</a> 。                                                | <ul style="list-style-type: none"> <li>调用批量数据迁移或实时数据接入API时无需配置。</li> <li>调用数据开发API时，有多个DataArts Studio实例时必选。只有一个DataArts实例时可选。不填写则默认查询default工作空间数据，如需查询其他工作空间需要带上该消息头。</li> <li>调用其他组件API时必选，否则会调用失败。</li> </ul> | d1cd7861478748a6925bc02f47c69279         |

| 名称         | 描述                           | 是否必选        | 示例                            |
|------------|------------------------------|-------------|-------------------------------|
| X-Dlm-Type | 数据服务版本类型，用于请求数据服务时，指定共享版/专享版 | 调用数据服务接口时必选 | 共享版: SHARED<br>专享版: EXCLUSIVE |

## 公共响应消息头

公共响应消息头如表14-2所示。

表 14-2 公共响应消息头

| 参数名            | 说明                  |
|----------------|---------------------|
| Content-Length | 响应消息体的字节长度，单位为Byte。 |
| Date           | 系统响应的时间。            |
| Content-type   | 发送的实体的MIME类型。       |

## 14.2 解析响应消息体中的流

导出作业和导出连接的接口响应消息是一个流，需要转化为一个文件。

您可以参考下面的样例代码：

```
String EXPORT_JOB_URL = "https://{endpoint}/v1/{project_id}/jobs/{job_name}/export";

try (CloseableHttpClient httpClient = HttpClients.createDefault()) {
    HttpPost httpPost = new HttpPost(EXPORT_JOB_URL);
    httpPost.setHeader("Content-Type", "application/json; charset=UTF-8");
    httpPost.setHeader("Accept", "application/octet-stream");
    httpPost.setHeader("X-Auth-Token", token);

    HttpResponse response = httpClient.execute(httpPost);
    int statusCode = response.getStatusLine().getStatusCode();
    if (statusCode == 200) {
        String filePath = "d:";
        String fileName = "job.zip";
        FileOutputStream fileOutputStream = new FileOutputStream(filePath + fileName);
        response.getEntity().writeTo(fileOutputStream);
    } else {
        System.out.println(statusCode);
    }
}
```

## 14.3 状态码

状态代码由三位数字组成，第一个数字定义了响应的类别，有五种可能取值：

- 1xx：指示信息，表示请求已接收，继续处理。
- 2xx：成功，表示请求已被成功接收、理解、接受。
- 3xx：重定向，要完成请求必须进行更进一步的操作。

- 4xx: 客户端错误, 请求有语法错误或请求无法实现。
- 5xx: 服务器端错误, 服务器未能实现合法的请求。

状态码如表14-3所示。

表 14-3 状态码

| 状态码 | 编码                            | 状态说明                                                    |
|-----|-------------------------------|---------------------------------------------------------|
| 100 | Continue                      | 继续请求。<br>这个临时响应用来通知客户端, 它的部分请求已经被服务器接收, 且仍未被拒绝。         |
| 101 | Switching Protocols           | 切换协议。只能切换到更高级的协议。<br>例如, 切换到HTTP的新版本协议。                 |
| 200 | OK                            | 接口调用成功。                                                 |
| 201 | Created                       | 创建类的请求完全成功。                                             |
| 202 | Accepted                      | 已经接受请求, 但未处理完成。                                         |
| 203 | Non-Authoritative Information | 非授权信息, 请求成功。                                            |
| 204 | NoContent                     | 请求完全成功, 同时HTTP响应不包含响应体。<br>在响应OPTIONS方法的HTTPS请求时返回此状态码。 |
| 205 | Reset Content                 | 重置内容, 服务器处理成功。                                          |
| 206 | Partial Content               | 服务器成功处理了部分GET请求。                                        |
| 300 | Multiple Choices              | 多种选择。请求的资源可包括多个位置, 相应可返回一个资源特征与地址的列表用于用户终端(例如: 浏览器)选择。  |
| 301 | Moved Permanently             | 永久移动, 请求的资源已被永久的移动到新的URI, 返回信息会包括新的URI。                 |
| 302 | Found                         | 资源被临时移动。                                                |
| 303 | See Other                     | 查看其它地址。使用GET和POST请求查看。                                  |
| 304 | Not Modified                  | 所请求的资源未修改, 服务器返回此状态码时, 不会返回任何资源。                        |
| 305 | Use Proxy                     | 所请求的资源必须通过代理访问。                                         |
| 306 | Unused                        | 已经被废弃的HTTP状态码。                                          |
| 400 | BadRequest                    | 非法请求。<br>建议直接修改该请求, 不要重试该请求。                            |

| 状态码 | 编码                            | 状态说明                                                                                    |
|-----|-------------------------------|-----------------------------------------------------------------------------------------|
| 401 | Unauthorized                  | 在客户端提供认证信息后，返回该状态码，表明服务端指出客户端所提供的认证信息不正确或非法。                                            |
| 402 | Payment Required              | 保留请求。                                                                                   |
| 403 | Forbidden                     | 请求被拒绝访问。<br>返回该状态码，表明请求能够到达服务端，且服务端能够理解用户请求，但是拒绝做更多的事情，因为该请求被设置为拒绝访问，建议直接修改该请求，不要重试该请求。 |
| 404 | NotFound                      | 所请求的资源不存在。<br>建议直接修改该请求，不要重试该请求。                                                        |
| 405 | MethodNotAllowed              | 请求中带有该资源不支持的方法。<br>建议直接修改该请求，不要重试该请求。                                                   |
| 406 | Not Acceptable                | 服务器无法根据客户端请求的内容特性完成请求。                                                                  |
| 407 | Proxy Authentication Required | 请求要求代理的身份认证，与401类似，但请求者应当使用代理进行授权。                                                      |
| 408 | Request Time-out              | 服务器等候请求时发生超时。<br>客户端可以随时再次提交该请求而无需进行任何更改。                                               |
| 409 | Conflict                      | 服务器在完成请求时发生冲突。<br>返回该状态码，表明客户端尝试创建的资源已经存在，或者由于冲突请求的更新操作不能被完成。                           |
| 410 | Gone                          | 客户端请求的资源已经不存在。返回该状态码，表明请求的资源已被永久删除。                                                     |
| 411 | Length Required               | 服务器无法处理客户端发送的不带Content-Length的请求信息。                                                     |
| 412 | Precondition Failed           | 未满足前提条件，服务器未满足请求者在请求中设置的其中一个前提条件。                                                       |
| 413 | Request Entity Too Large      | 由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个Retry-After的响应信息。   |
| 414 | Request-URI Too Large         | 请求的URI过长（URI通常为网址），服务器无法处理。                                                             |

| 状态码 | 编码                              | 状态说明                                                                         |
|-----|---------------------------------|------------------------------------------------------------------------------|
| 415 | Unsupported Media Type          | 服务器无法处理请求附带的媒体格式。                                                            |
| 416 | Requested range not satisfiable | 客户端请求的范围无效。                                                                  |
| 417 | Expectation Failed              | 服务器无法满足Expect的请求头信息。                                                         |
| 422 | Unprocessable Entity            | 请求格式正确，但是由于含有语义错误，无法响应。                                                      |
| 429 | TooManyRequests                 | 表明请求超出了客户端访问频率的限制或者服务端接收到多于它能处理的请求。建议客户端读取相应的Retry-After首部，然后等待该首部指出的时间后再重试。 |
| 500 | InternalServerError             | 表明服务端能被请求访问到，但是不能理解用户的请求。                                                    |
| 501 | Not Implemented                 | 服务器不支持请求的功能，无法完成请求。                                                          |
| 502 | Bad Gateway                     | 充当网关或代理的服务器，从远端服务器接收到了一个无效的请求。                                               |
| 503 | ServiceUnavailable              | 被请求的服务无效。<br>建议直接修改该请求，不要重试该请求。                                              |
| 504 | ServerTimeout                   | 请求在给定的时间内无法完成。客户端仅在为请求指定超时（Timeout）参数时会得到该响应。                                |
| 505 | HTTP Version not supported      | 服务器不支持请求的HTTP协议的版本，无法完成处理。                                                   |

## 14.4 错误码

### 14.4.1 数据集成 API 错误码

调用API出错后，将不会返回结果数据。调用方可根据每个API对应的错误码来定位错误原因。当调用出错时，HTTP请求返回一个4xx或5xx的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。在调用方找不到错误原因时，可以联系客服，并提供错误码，以便尽快帮您解决问题。

- 异常响应样例

```
{
  "errCode": "Cdm.0100",
  "externalMessage": "Job[jdbc2hive] doesn't exist."
}
```

- 参数说明

| 参数              | 是否必选 | 类型     | 说明    |
|-----------------|------|--------|-------|
| errCode         | 否    | String | 错误码。  |
| externalMessage | 否    | String | 错误消息。 |

- 错误码说明

以下错误信息中的 %s 为变量，实际返回信息时会替换为具体的参数名、表名、作业名、连接名等。

当您调用 API 时，如果遇到“APIGW”开头的错误码，请参见 [API 网关错误码](#) 进行处理。

| 错误码      | 状态码 | 错误信息                                   | 描述                     | 处理措施                             |
|----------|-----|----------------------------------------|------------------------|----------------------------------|
| Cdm.0000 | 400 | 系统错误。                                  | 系统错误。                  | 请联系客服或技术支持人员协助解决。                |
| Cdm.0001 | 400 | 资源不存在或不合法。                             | 请求的资源不存在或无访问权限。        | 请联系客服或技术支持人员协助解决。                |
| Cdm.0004 | 400 | 无效的参数类型                                | 输入参数和类型不匹配             | 请根据错误提示将参数修改正确后请重试。              |
| Cdm.0009 | 400 | %s 不是整型数字或超出整型数的取值范围 [0 ~ 2147483647]。 | 输入参数不是整型数字或超出整型数的取值范围。 | 请根据错误提示将参数修改正确后请重试。              |
| Cdm.0010 | 400 | 整数必须在区间 [%s]。                          | 校验程序的参数缺失或长度为 0。       | 请根据错误提示将参数修改正确后请重试。              |
| Cdm.0011 | 400 | 输入超过取值范围。                              | 参数格式不正确或超过取值范围，无法解析。   | 请根据返回的详细错误信息，确认参数值是否合法，修改正确后请重试。 |
| Cdm.0012 | 400 | 没有匹配的数据库 JDBC 驱动。                      | 没有匹配的数据库 JDBC 驱动。      | 请联系客服或技术支持人员协助解决。                |
| Cdm.0014 | 400 | 非法参数。                                  | 参数不合法。                 | 请确认参数值是否合法，修改正确后请重试。             |

| 错误码      | 状态码 | 错误信息                     | 描述                 | 处理措施                                                                                                                                   |
|----------|-----|--------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Cdm.0015 | 400 | 解析文件内容出错。                | 解析文件内容失败。          | 请确认上传的文件内容或格式是否正确，修改正确后请重试。                                                                                                            |
| Cdm.0016 | 400 | 上传文件不能为空。                | 上传的文件为空。           | 请确认上传的文件是否为空，修改正确后请重试。                                                                                                                 |
| Cdm.0017 | 400 | 无法将输入值保存到存储库             | 无法将输入值保存到存储库。      | 请联系客服或技术支持人员协助解决。                                                                                                                      |
| Cdm.0018 | 400 | 作业和连接内容不合法。              | 作业和连接内容非法。         | 请联系客服或技术支持人员协助解决。                                                                                                                      |
| Cdm.0019 | 400 | 无法删除存储库中的链接              | 删除存储库中的链接失败。       | 请稍后重试，或联系或客服或技术支持人员协助解决。                                                                                                               |
| Cdm.0020 | 400 | 必须包含子字符串： %s。            | 被校验参数为空或不包含指定子字符串。 | 请根据错误提示将参数修改正确后，再重试。                                                                                                                   |
| Cdm.0021 | 400 | 不能连接服务器： %s。             | 连接服务器失败。           | 请联系客服或技术支持人员协助解决。                                                                                                                      |
| Cdm.0024 | 400 | [ %s]必须在区间[ %s]。         | 被校验参数不在指定区间范围内。    | 请根据错误提示将参数修改正确后，再重试。                                                                                                                   |
| Cdm.0031 | 400 | 无法创建新的提交数据               | 无法创建新的提交数据         | 请联系客服或技术支持人员协助解决。                                                                                                                      |
| Cdm.0032 | 400 | 当前用户没有操作权限，请通过IAM检查账户权限！ | 用户权限不足             | 在CDM控制台操作时，请参考 <a href="#">CDM权限管理</a> 为该用户授予足够的操作权限。<br>在DataArts Studio控制台操作时，请参考 <a href="#">DataArts Studio权限管理</a> 为该用户授予足够的操作权限。 |



| 错误码      | 状态码 | 错误信息                      | 描述                                                                      | 处理措施                                         |
|----------|-----|---------------------------|-------------------------------------------------------------------------|----------------------------------------------|
| Cdm.0036 | 400 | 内部错误。                     | 内部错误。<br>该报错由内部错误引起，显示的报错信息可能不唯一，请根据真实报错（如“内部错误：空指针异常”）联系客服或技术支持人员协助解决。 | 请联系客服或技术支持人员协助解决。                            |
| Cdm.0037 | 400 | 无法提交作业。                   | 无法提交作业。                                                                 | 请联系客服或技术支持人员协助解决。                            |
| Cdm.0051 | 400 | 无效的提交引擎： <i>%s</i> 。      | 作业引擎名称非法。                                                               | 请指定正确的作业引擎后再重试。                              |
| Cdm.0052 | 400 | 作业 <i>%s</i> 正在运行。        | 作业正在运行。                                                                 | 作业正在运行，无法执行当前操作，请等待作业运行结束后再重试。               |
| Cdm.0053 | 400 | 作业 <i>%s</i> 未运行。         | 作业未运行。                                                                  | 请运行作业后再重试。                                   |
| Cdm.0054 | 400 | 作业 <i>%s</i> 不存在。         | 作业不存在。                                                                  | 请确认作业是否存在。                                   |
| Cdm.0055 | 400 | 作业类型不支持。                  | 作业类型不支持。                                                                | 请参考官网，配置支持的作业类型。                             |
| Cdm.0056 | 400 | 不能提交作业。原因： <i>%s</i> 。    | 作业提交失败。                                                                 | 请根据返回的详细错误信息，定位原因，修改正确后请重试。                  |
| Cdm.0057 | 400 | 无效的作业执行引擎： <i>%s</i> 。    | 作业引擎无效。                                                                 | 请指定正确的作业引擎后再重试。                              |
| Cdm.0058 | 400 | 提交和执行引擎组合不合法。             | 提交和执行引擎组合不合法。                                                           | 请指定正确的作业引擎后再重试。                              |
| Cdm.0059 | 400 | 作业 <i>%s</i> 已被禁用。不能提交作业。 | 作业已被禁用，无法提交。                                                            | 当前作业无法提交，建议重新创建一个作业后再重试。或者，请联系客服或技术支持人员协助解决。 |

| 错误码      | 状态码 | 错误信息                            | 描述                 | 处理措施                                |
|----------|-----|---------------------------------|--------------------|-------------------------------------|
| Cdm.0060 | 400 | 作业使用的连接 %s 已被禁用。不能提交作业。         | 作业使用的连接已被禁用。       | 请改为其他连接后，再重新提交作业。                   |
| Cdm.0061 | 400 | 连接器 %s 不支持此方向。不能提交作业。           | 该连接器不能作为作业的源端或目的端。 | 该连接器不能作为作业的源端或目的端，请改为其他连接后，再重新提交作业。 |
| Cdm.0062 | 400 | 二进制文件仅适合 SFTP/FTP/HDFS/OBS 连接器。 | 连接器不在指定的范围内。       | 请指定正确的连接器后再重试。                      |
| Cdm.0063 | 400 | 创建表格错误。原因： %s。                  | 创建表格失败。            | 请根据返回的详细错误信息定位原因，修改正确后请重试。          |
| Cdm.0064 | 400 | 数据格式不匹配。                        | 数据格式不匹配。           | 请根据返回的详细错误信息，确认数据格式是否正确，修改正确后请重试。   |
| Cdm.0065 | 400 | 定时器启动失败，原因 %s。                  | 定时器启动失败。           | 请联系客服或技术支持人员协助解决。                   |
| Cdm.0066 | 400 | 获取样值失败，原因： %s。                  | 获取样值失败。            | 请联系客服或技术支持人员协助解决。                   |
| Cdm.0067 | 400 | 获取Schema失败，原因： %s。              | 获取Schema失败。        | 请联系客服或技术支持人员协助解决。                   |
| Cdm.0085 | 400 | %s 超过最大值 %s。                    | 参数超过最大值。           | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.0089 | 400 | 配置项 [%s] 不存在。                   | 配置项不存在。            | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |

| 错误码      | 状态码 | 错误信息                | 描述           | 处理措施                                               |
|----------|-----|---------------------|--------------|----------------------------------------------------|
| Cdm.0100 | 400 | 作业[%s]不存在。          | 作业不存在。       | 请指定正确的作业后再重试。                                      |
| Cdm.0101 | 400 | 连接[%s]不存在。          | 连接不存在。       | 请指定正确的连接后再重试。                                      |
| Cdm.0102 | 400 | 连接器[%s]不存在。         | 连接器不存在。      | 请指定正确的连接器后再重试。                                     |
| Cdm.0104 | 400 | 作业名已存在。             | 作业名已存在。      | 作业名已存在，请重新命名后，再重试。                                 |
| Cdm.0201 | 400 | 获取实例失败。             | 获取实例失败。      | 请联系客服或技术支持人员协助解决。                                  |
| Cdm.0202 | 400 | 作业状态未知。             | 作业状态未知。      | 请稍后重试，或请联系客服或技术支持人员协助解决。                           |
| Cdm.0204 | 400 | 没有已创建的MRS连接。        | 没有已创建的MRS连接。 | 当前没有MRS连接，您需要先前往集群的“连接管理”页面创建一个MRS连接，然后再重新执行当前的操作。 |
| Cdm.0230 | 400 | 不能加载该类：%s。          | 类加载失败。       | 请联系客服或技术支持人员协助解决。                                  |
| Cdm.0231 | 400 | 不能初始化该类：%s。         | 类初始化失败。      | 请联系客服或技术支持人员协助解决。                                  |
| Cdm.0232 | 400 | 数据写入失败。原因：<br>%s。   | 数据写入失败。      | 请联系客服或技术支持人员协助解决。                                  |
| Cdm.0233 | 400 | 提取数据过程异常。原因：<br>%s。 | 提取数据过程异常。    | 请联系客服或技术支持人员协助解决。                                  |
| Cdm.0234 | 400 | 载入数据过程异常。原因：<br>%s。 | 载入数据过程异常。    | 请联系客服或技术支持人员协助解决。                                  |

| 错误码      | 状态码 | 错误信息              | 描述              | 处理措施                                        |
|----------|-----|-------------------|-----------------|---------------------------------------------|
| Cdm.0235 | 400 | 数据已全部消费完毕。原因: %s。 | 数据已全部消费完毕。      | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。       |
| Cdm.0236 | 400 | 从分区程序中检索到无效分区数。   | 从分区程序中检索到无效分区数。 | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。       |
| Cdm.0238 | 400 | %s不能为空。           | 参数不合法。          | 请根据错误提示将参数修改正确后再重试。                         |
| Cdm.0240 | 400 | 获取文件 %s 状态失败。     | 获取文件状态失败。       | 请联系客服或技术支持人员协助解决。                           |
| Cdm.0241 | 400 | 获取文件 %s 类型失败。     | 获取文件类型失败。       | 请联系客服或技术支持人员协助解决。                           |
| Cdm.0242 | 400 | 文件检查异常: %s。       | 文件检查异常。         | 请联系客服或技术支持人员协助解决。                           |
| Cdm.0243 | 400 | 重命名 %s 为 %s 失败。   | 重命名失败。          | 可能是名称已存在, 请重新命名后再重试。                        |
| Cdm.0244 | 400 | 创建文件 %s 失败。       | 创建文件失败。         | 请确认是否具有创建权限, 或稍后重试。若无法解决, 请联系客服或技术支持人员协助解决。 |
| Cdm.0245 | 400 | 删除文件 %s 失败。       | 删除文件失败。         | 请确认是否具有删除权限, 或稍后重试。若无法解决, 请联系客服或技术支持人员协助解决。 |
| Cdm.0246 | 400 | 创建目录 %s 失败。       | 创建目录失败。         | 请确认是否具有创建权限, 或稍后重试。若无法解决, 请联系客服或技术支持人员协助解决。 |

| 错误码      | 状态码 | 错误信息                     | 描述             | 处理措施                                          |
|----------|-----|--------------------------|----------------|-----------------------------------------------|
| Cdm.0247 | 400 | 操作HBase失败。原因：<br>%s。     | 操作HBase失败。     | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。           |
| Cdm.0248 | 400 | 清空%s数据失败。原因：<br>%s。      | 清空数据失败。        | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。           |
| Cdm.0249 | 400 | 文件名%s无效。                 | 文件名无效。         | 请将文件名修改正确后，再重试。                               |
| Cdm.0250 | 400 | 不能操作该路径：<br>%s。          | 不能操作该路径。       | 请确认是否具有该路径的操作权限，或稍后重试。若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.0251 | 400 | 向HBase加载数据失败。原因：<br>%s。  | 向HBase加载数据失败。  | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。           |
| Cdm.0307 | 400 | 无法获得请求事务的连接租约，原因：<br>%s。 | 无法获得请求事务的连接租约。 | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。           |
| Cdm.0315 | 400 | 连接名%s已存在。                | 该连接已存在。        | 请指定其他连接名后再重试。                                 |
| Cdm.0316 | 400 | 无法更新不存在的连接。              | 无法更新不存在的连接。    | 请指定正确的连接后再重试。                                 |
| Cdm.0317 | 400 | 连接%s无效。                  | 连接无效。          | 请指定正确的连接后再重试。                                 |
| Cdm.0318 | 400 | 作业已存在，无法重复创建。            | 作业已存在。         | 请指定其他作业名再重试。                                  |
| Cdm.0319 | 400 | 无法更新不存在的作业。              | 无法更新不存在的作业。    | 请确认待更新的作业是否存在，作业名修改正确后再重试。                    |

| 错误码      | 状态码 | 错误信息           | 描述             | 处理措施                                  |
|----------|-----|----------------|----------------|---------------------------------------|
| Cdm.0320 | 400 | 作业 %s 无效。      | 作业无效。          | 请联系客服或技术支持人员协助解决。                     |
| Cdm.0321 | 400 | 连接 %s 已被使用。    | 连接已被使用。        | 连接已被使用，无法执行当前的操作，请将连接释放后再重试。          |
| Cdm.0322 | 400 | 作业 %s 已被使用。    | 作业已被使用。        | 请联系客服或技术支持人员协助解决。                     |
| Cdm.0323 | 400 | 该提交已存在，无法重复创建。 | 该提交已存在。        | 您已提交过相同操作的请求，请稍后再重试。                  |
| Cdm.0327 | 400 | 无效的连接或作业： %s。  | 无效的连接或作业。      | 请指定正确的连接或作业再重试。                       |
| Cdm.0411 | 400 | 连接到文件服务器时出错。   | 连接到文件服务器时出错。   | 请联系客服或技术支持人员协助解决。                     |
| Cdm.0413 | 400 | 向文件服务器传输数据时出错。 | 向文件服务器传输数据时出错。 | 请联系客服或技术支持人员协助解决。                     |
| Cdm.0415 | 400 | 从文件服务器下载文件出错。  | 从文件服务器下载文件出错。  | 请联系客服或技术支持人员协助解决。                     |
| Cdm.0416 | 400 | 抽取数据时出错。       | 抽取数据时出错。       | 请联系客服或技术支持人员协助解决。                     |
| Cdm.0420 | 400 | 源文件或源目录不存在。    | 源文件或源目录不存在。    | 请确认源文件或源目录是否存在，修改正确后再重试。              |
| Cdm.0423 | 400 | 目的路径存在重复文件。    | 目的路径存在重复文件。    | 请在目的路径中删除重复文件后再重试。                    |
| Cdm.0467 | 400 | 连接超时。          | 连接超时。          | 请检查IP、主机名、端口填写是否正确，检查网络安全组和防火墙配置是否正确。 |
| Cdm.0501 | 400 | 无效的URI[ %s]。   | 无效的URI。        | 请指定正确的URI后，再重试。                       |

| 错误码      | 状态码 | 错误信息                    | 描述            | 处理措施                                                                                           |
|----------|-----|-------------------------|---------------|------------------------------------------------------------------------------------------------|
| Cdm.0518 | 400 | 连接HDFS失败。原因：<br>%s。     | 连接HDFS失败。     | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。                                                            |
| Cdm.0600 | 400 | 无法连接FTP服务器。             | 无法连接FTP服务器。   | 可能是由于网络不通、安全组或防火墙规则未放行、FTP主机名无法解析、FTP用户名密码错误等原因。若排除上述原因后仍无法解决，请联系客服或技术支持人员协助解决。                |
| Cdm.0700 | 400 | 无法连接SFTP服务器。            | 无法连接SFTP服务器。  | 可能是由于网络不通、安全组或防火墙规则未放行、SFTP主机名无法解析、SFTP用户名密码错误等原因。若排除上述原因后仍无法解决，请联系客服或技术支持人员协助解决。              |
| Cdm.0800 | 400 | 无法连接OBS服务器。             | 无法连接OBS服务器。   | 可能是由于OBS终端节点与当前区域不一致、AK/SK错误、AK/SK不是当前用户的AK/SK、安全组或防火墙规则未放行等原因。若排除上述原因后仍无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.0801 | 400 | OBS桶[%s]不存在。            | OBS桶不存在。      | 指定的OBS桶可能不存在或不在当前区域，请指定正确的OBS桶后再重试。                                                            |
| Cdm.0831 | 400 | 无法连接到KODO服务器。原因：<br>%s。 | 无法连接到KODO服务器。 | 请联系客服或技术支持人员协助解决。                                                                              |

| 错误码      | 状态码 | 错误信息                  | 描述                 | 处理措施                                                              |
|----------|-----|-----------------------|--------------------|-------------------------------------------------------------------|
| Cdm.0900 | 400 | 表[%s]不存在。             | 表不存在。              | 请指定正确的表名后再重试。                                                     |
| Cdm.0901 | 400 | 无法连接数据库服务器。原因: %s。    | 无法连接数据库服务器。        | 请联系客服或技术支持人员协助解决。                                                 |
| Cdm.0902 | 400 | SQL语句无法执行。原因 %s。      | SQL语句无法执行。         | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。                             |
| Cdm.0903 | 400 | 元数据获取失败。原因: %s。       | 元数据获取失败。           | 请确认在集群的“连接管理”页面创建连接时引用符号是否正确或查看数据库表是否存在。若仍无法解决, 请联系客服或技术支持人员协助解决。 |
| Cdm.0904 | 400 | 从结果中检索数据时发生错误。原因: %s。 | 从结果中检索数据时发生错误。     | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。                             |
| Cdm.0913 | 400 | Schema和SQL不可以同时为空。    | Schema和SQL需指定其中一项。 | 请确认Schema和SQL是否同时为空, 请指定其中一项后, 再重试。                               |
| Cdm.0916 | 400 | 增量读取情况下必须指定上次的值。      | 增量读取时未指定上次的值。      | 请指定上次的值后再重试。                                                      |
| Cdm.0917 | 400 | 缺少字段检查将无法获得上次的值。      | 缺少字段。              | 请联系客服或技术支持人员协助解决。                                                 |
| Cdm.0921 | 400 | 不支持类型 %s。             | 类型不合法。             | 请指定正确的类型后再重试。                                                     |
| Cdm.0925 | 400 | 分区字段含有不支持的值。          | 分区字段含有不支持的值。       | 请确认分区字段是否含有不支持的值, 修改正确后再重试。                                       |



| 错误码      | 状态码 | 错误信息                            | 描述                      | 处理措施                                                                           |
|----------|-----|---------------------------------|-------------------------|--------------------------------------------------------------------------------|
| Cdm.0926 | 400 | 取不到 Schema。原因: %s。              | 获取Schema失败。             | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。                                          |
| Cdm.0927 | 400 | 中转表不为空。                         | 中转表不为空。                 | 请指定一个空的中转表后再重试。                                                                |
| Cdm.0928 | 400 | 中转表到目的表进行数据迁移时发生错误。             | 中转表到目的表进行数据迁移时发生错误。     | 请联系客服或技术支持人员协助解决。                                                              |
| Cdm.0931 | 400 | Schema字段大小[%s]与结果集的字段大小[%s]不匹配。 | Schema字段大小与结果集的字段大小不匹配。 | 请将Schema字段大小和结果集中的字段大小改为一致后再重试。                                                |
| Cdm.0932 | 400 | 找不到字段最大值。                       | 找不到字段最大值。               | 请联系客服或技术支持人员协助解决。                                                              |
| Cdm.0934 | 400 | 不同Schema/Catalog下有重名表。          | 不同Schema/Catalog下有重名表。  | 请联系客服或技术支持人员协助解决。                                                              |
| Cdm.0936 | 400 | 错误脏数据条数达到上限。                    | 错误脏数据条数达到上限。            | 您可以编辑作业, 在作业的任务配置中将错误脏数据条数增大。                                                  |
| Cdm.0940 | 400 | 表名准确匹配失败。                       | 表名准确匹配失败。               | 匹配不到表名, 请指定正确的表名后再重试。                                                          |
| Cdm.0941 | 400 | 无法连接服务器。原因: [%s]                | 无法连接服务器。                | 请检查IP、主机名、端口填写是否正确, 检查网络安全组和防火墙配置是否正确, 参考数据库返回消息进行定位。若仍无法解决, 请联系客服或技术支持人员协助解决。 |
| Cdm.0950 | 400 | 当前认证信息无法连接到数据库。                 | 当前认证信息无法连接到数据库。         | 认证信息错误, 请修改正确后再重试。                                                             |

| 错误码      | 状态码 | 错误信息                         | 描述        | 处理措施                                                                                                                                                                                                               |
|----------|-----|------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cdm.0962 | 400 | 必须指定主机IP。                    | 未指定主机IP。  | 未指定主机IP, 请指定主机IP后, 再重试。                                                                                                                                                                                            |
| Cdm.0963 | 400 | 必须指定主机端口。                    | 未指定主机端口。  | 未指定主机端口, 请指定主机端口后, 再重试。                                                                                                                                                                                            |
| Cdm.0964 | 400 | 必须指定数据库。                     | 未指定数据库。   | 未指定数据库, 请指定数据库后, 再重试。                                                                                                                                                                                              |
| Cdm.1000 | 400 | Hive表[%s]不存在。                | Hive表不存在。 | 请输入正确的Hive表名后, 再重试。                                                                                                                                                                                                |
| Cdm.1010 | 400 | 无效的URI %s。URI必须为null或有效的URI。 | 无效的URI。   | 请输入正确的URI后, 再重试。下面是一些URI示例: <ul style="list-style-type: none"> <li>• hdfs://example.com:8020/</li> <li>• hdfs://example.com/</li> <li>• file:///</li> <li>• file:///tmp</li> <li>• file://localhost/tmp</li> </ul> |
| Cdm.1011 | 400 | 连接Hive失败, 原因: %s。            | 连接Hive失败。 | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。                                                                                                                                                                              |
| Cdm.1100 | 400 | 表[%s]不存在。                    | 表不存在。     | 请确认表是否存在, 输入正确的表名后再重试。                                                                                                                                                                                             |
| Cdm.1101 | 400 | 获取连接失败, 原因: %s。              | 获取连接失败。   | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。                                                                                                                                                                              |
| Cdm.1102 | 400 | 创表失败, 原因: %s。                | 创建表失败。    | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。                                                                                                                                                                              |

| 错误码      | 状态码 | 错误信息                            | 描述                              | 处理措施                                |
|----------|-----|---------------------------------|---------------------------------|-------------------------------------|
| Cdm.1103 | 400 | 未设置 Rowkey。                     | 未设置 Rowkey。                     | 请设置 Rowkey 后再重试。                    |
| Cdm.1104 | 400 | 打开表格失败。原因：<br>%s。               | 打开表格失败。                         | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1105 | 400 | 作业初始化失败。原因 %s。                  | 作业初始化失败。                        | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1111 | 400 | 表名不能为空。                         | 表名为空。                           | 请输入正确的表名后，再重试。                      |
| Cdm.1114 | 400 | Rowkey 为空，请在字段映射步骤重新设置。         | Rowkey 为空。                      | 请按照错误提示进行处理。                        |
| Cdm.1115 | 400 | Columns 为空，请在字段映射步骤重新设置。        | Columns 为空。                     | 请按照错误提示进行处理。                        |
| Cdm.1116 | 400 | 列名重复，请在字段映射步骤重新设置。              | 列名重复。                           | 请按照错误提示进行处理。                        |
| Cdm.1117 | 400 | 判断表格是否存在失败，原因：<br>%s。           | 判断表格是否存在失败。                     | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1118 | 400 | 表 %s 不包含列族 %s。                  | 表中不包含列族。                        | 请指定列族后再重试。                          |
| Cdm.1120 | 400 | 表中有数据，请清空表数据或重新设置导入前是否清空表数据配置项。 | 表中有数据，请清空表数据或重新设置导入前是否清空表数据配置项。 | 请按照错误提示进行处理。                        |

| 错误码      | 状态码 | 错误信息                            | 描述                              | 处理措施                                |
|----------|-----|---------------------------------|---------------------------------|-------------------------------------|
| Cdm.1121 | 400 | 关闭连接已失败。原因：<br>%s。              | 关闭连接失败。                         | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1201 | 400 | 不能连接到Redis服务器，原因：<br>%s。        | 无法连接到Redis服务器。                  | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1203 | 400 | 从Redis服务器抽取数据失败，原因：<br>%s。      | 从Redis服务器获取数据失败。                | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1205 | 400 | Redis值前缀不能为空白符。                 | Redis值前缀不能为空白符。                 | 请去除Redis前缀前的空白符，然后再重试。              |
| Cdm.1206 | 400 | Redis值存储类型必须指定为“string”或“hash”。 | Redis值存储类型必须指定为“string”或“hash”。 | 请按照错误提示进行处理。                        |
| Cdm.1207 | 400 | 当值存储类型为“string”时，必须指定值分隔符。      | 值存储类型为“string”，未指定值分隔符。         | 请指定分隔符后再重试。                         |
| Cdm.1208 | 400 | Redis存储字段列表必须指定。                | Redis存储字段列表未指定。                 | 请指定Redis存储字段列表后再重试。                 |
| Cdm.1209 | 400 | Redis键分隔符不能为空白符。                | Redis键分隔符不能为空白符。                | 请输入正确的分隔符后，再重试。                     |
| Cdm.1210 | 400 | 必须指定Redis主键字段列表。                | Redis主键字段列表未指定。                 | 请指定Redis主键字段列表后再重试。                 |
| Cdm.1211 | 400 | Redis主键字段列表必须在字段列表中存在。          | Redis主键字段列表不在字段列表中。             | 请指定Redis主键字段列表后再重试。                 |
| Cdm.1213 | 400 | 必须指定Redis服务器列表。                 | 未指定Redis服务器列表。                  | 请指定Redis服务器列表后再重试。                  |

| 错误码      | 状态码 | 错误信息                                    | 描述                       | 处理措施                                |
|----------|-----|-----------------------------------------|--------------------------|-------------------------------------|
| Cdm.1301 | 400 | 不能连接到MongoDB服务器，原因： <i>%s</i> 。         | 连接到MongoDB服务器失败          | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1302 | 400 | 从MongoDB服务器抽取数据失败，原因： <i>%s</i> 。       | 从MongoDB服务器抽取数据失败。       | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1304 | 400 | 必须指定MongoDB服务器的集合。                      | 未指定MongoDB服务器的集合。        | 未指定MongoDB服务器的集合，请指定后，再重试。          |
| Cdm.1305 | 400 | 必须指定MongoDB服务列表。                        | 未指定MongoDB服务列表。          | 未指定MongoDB服务列表，请指定后，再重试。            |
| Cdm.1306 | 400 | 必须指定MongoDB服务的数据库名称。                    | 未指定MongoDB服务的数据库名称。      | 未指定MongoDB服务的数据库名称，请指定数据库后，再重试。     |
| Cdm.1307 | 400 | 必须指定MongoDB服务的字段列表。                     | 未指定MongoDB服务的字段列表。       | 未指定MongoDB服务的字段列表，请指定字段列表后，再重试。     |
| Cdm.1501 | 400 | 不能连接到Elasticsearch服务器，原因： <i>%s</i> 。   | 无法连接到Elasticsearch服务器    | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1502 | 400 | 向Elasticsearch服务器写入数据失败，原因： <i>%s</i> 。 | 向Elasticsearch服务器写入数据失败。 | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |
| Cdm.1503 | 400 | 关闭Elasticsearch连接失败，原因： <i>%s</i> 。     | 关闭Elasticsearch连接失败。     | 请根据错误提示进行定位，若无法解决，请联系客服或技术支持人员协助解决。 |

| 错误码      | 状态码 | 错误信息                           | 描述                           | 处理措施                                    |
|----------|-----|--------------------------------|------------------------------|-----------------------------------------|
| Cdm.1504 | 400 | 获取Elasticsearch索引错误, 原因: %s。   | 获取Elasticsearch索引错误          | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。   |
| Cdm.1505 | 400 | 获取Elasticsearch类型错误, 原因: %s。   | 获取Elasticsearch类型错误。         | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。   |
| Cdm.1506 | 400 | 获取Elasticsearch文档字段错误, 原因: %s。 | 获取Elasticsearch文档字段错误。       | 请根据错误提示进行定位, 若无法解决, 请联系客服或技术支持人员协助解决。   |
| Cdm.1508 | 400 | 必须指定Elasticsearch服务器主机名或IP地址。  | 未指定Elasticsearch服务器主机名或IP地址。 | 未指定Elasticsearch服务器主机名或IP地址, 请指定后, 再重试。 |
| Cdm.1510 | 400 | 必须指定Elasticsearch索引。           | 未指定Elasticsearch索引。          | 当前未指定Elasticsearch索引, 请指定后再重试。          |
| Cdm.1511 | 400 | 必须指定Elasticsearch类型。           | 未指定Elasticsearch类型。          | 当前未指定Elasticsearch类型, 请指定后再重试。          |
| Cdm.1513 | 400 | 字段列表中必须包含字段类型定义。               | 字段列表中未包含字段类型定义。              | 请确认字段列表中是否包含字段类型定义, 修改正确后再重试。           |
| Cdm.1514 | 400 | 字段列表中必须包含主键字段。                 | 未设置主键字段。                     | 当前未设置主键字段, 请设置主键字段后再重试。                 |
| Cdm.1516 | 400 | 非法列名 %s。                       | 列名不合法。                       | 请确认列名是否合法, 输入正确的列名后再重试。                 |
| Cdm.1517 | 400 | 获取文档数量产生错误。                    | 获取文档数量产生错误。                  | 请联系客服或技术支持人员协助解决。                       |
| Cdm.1519 | 400 | 抽取数据错误。                        | 抽取数据错误。                      | 请联系客服或技术支持人员协助解决。                       |

| 错误码      | 状态码 | 错误信息             | 描述           | 处理措施                                    |
|----------|-----|------------------|--------------|-----------------------------------------|
| Cdm.1601 | 400 | 连接服务器失败。         | 连接服务器失败。     | 请联系客服或技术支持人员协助解决。                       |
| Cdm.1603 | 400 | 获取topic %s的样值失败。 | 获取topic样值失败。 | 请联系客服或技术支持人员协助解决。                       |
| Cdm.1604 | 400 | topic %s没有数据。    | 该topic中无数据。  | 该topic中无数据，请排查无数据的原因。或者，请改为其他topic后再重试。 |

## 14.4.2 数据开发错误码

错误响应消息格式如下：

```
{
  "error_code": "DLF.0100",
  "error_msg": "The job not found"
}
```

错误码如表14-4所示。

表 14-4 错误码

| 状态码 | 错误码      | 错误信息                                        | 描述              | 处理措施         |
|-----|----------|---------------------------------------------|-----------------|--------------|
| 400 | DLF.0100 | The job does not exist.                     | 作业不存在。          | 请检查作业是否存在。   |
| 400 | DLF.0101 | The job quota has reached the upper limit.  | 作业配额已达上限。       | 请申请作业配额。     |
| 400 | DLF.0102 | The job name has been used by another job.  | 作业名称已经被其他作业占用。  | 请检查作业是否被占用。  |
| 400 | DLF.0136 | This API can be called only for batch jobs. | 只有批作业才支持这个接口调用。 | 请检查接口调用作业。   |
| 400 | DLF.0137 | The job instance does not exist.            | 作业实例不存在。        | 请检查作业实例是否存在。 |

| 状态码 | 错误码      | 错误信息                                             | 描述             | 处理措施          |
|-----|----------|--------------------------------------------------|----------------|---------------|
| 400 | DLF.0201 | The script fails to be created.                  | 创建脚本失败。        | 请检查脚本。        |
| 400 | DLF.0202 | The script name has been used by another script. | 脚本名称已经被其他脚本占用。 | 请检查脚本名称是否被占用。 |
| 400 | DLF.0203 | The script fails to be modified.                 | 修改脚本失败。        | 请检查脚本。        |
| 400 | DLF.0802 | The specified parameter is not configured.       | 指定的参数没有配置。     | 请检查参数。        |
| 400 | DLF.0803 | The OBS path is invalid.                         | OBS路径不合法。      | 请检查OBS路径。     |
| 400 | DLF.0810 | The queried task does not exist.                 | 查询的系统任务不存在。    | 请检查系统任务。      |
| 400 | DLF.0815 | The OBS file fails to be accessed.               | 读取OBS文件失败。     | 请检查OBS文件。     |
| 400 | DLF.1006 | The job node is empty.                           | 作业的节点为空。       | 请检查节点。        |
| 400 | DLF.1242 | The OBS bucket does not exist.                   | OBS桶不存在。       | 请检查OBS桶。      |
| 400 | DLF.2014 | The job object type is invalid.                  | 作业类型不合法。       | 请检查作业类型。      |
| 400 | DLF.3004 | The job name cannot be left blank.               | 作业名称不能为空。      | 请检查作业名称。      |
| 400 | DLF.3018 | The job name is invalid.                         | 作业名称不合法。       | 请检查作业名称。      |
| 400 | DLF.3025 | The job scheduling parameter is invalid.         | 作业的调度参数不合法。    | 请检查调度参数。      |
| 400 | DLF.3050 | The job description is invalid.                  | 作业的描述不合法。      | 请检查作业的描述。     |
| 400 | DLF.6201 | The script does not exist.                       | 脚本不存在。         | 请检查脚本。        |



| 状态码 | 错误码      | 错误信息                                                           | 描述               | 处理措施              |
|-----|----------|----------------------------------------------------------------|------------------|-------------------|
| 400 | DLF.6205 | The script execution instance does not exist.                  | 脚本的执行实例不存在。      | 请检查脚本的执行实例。       |
| 400 | DLF.6241 | The resource to be queried does not exist.                     | 查询的资源不存在。        | 请检查资源是否存在。        |
| 400 | DLF.6247 | The script type is not specified.                              | 脚本类型没有指定。        | 请检查脚本类型。          |
| 400 | DLF.6253 | The script quota has reached the upper limit.                  | 脚本配额已达上限。        | 请申请脚本配额。          |
| 400 | DLF.6258 | The directory contains data that cannot be deleted.            | 目录下有数据无法删除。      | 请先删除目录下数据。        |
| 400 | DLF.6259 | A resource with the same name already exists in the directory. | 目录下已经存在同名的资源。    | 请检查目录下是否存在同名的资源。  |
| 400 | DLF.6263 | The resource type is invalid.                                  | 资源类型不合法。         | 请检查资源类型。          |
| 400 | DLF.6264 | The OBS path of the resource file is invalid.                  | 资源文件所在的OBS路径不合法。 | 请检查OBS路径。         |
| 400 | DLF.6265 | The resource description is invalid.                           | 资源描述不合法。         | 请检查资源描述。          |
| 400 | DLF.6271 | Each user can create only one DLI connection.                  | 每个用户只能创建一个DLI连接。 | 请检查DLI连接。         |
| 400 | DLF.6309 | The connection name has been used by another connection.       | 连接名称已经在其他连接占用。   | 请检查连接名称是否被其他连接占用。 |
| 400 | DLF.6322 | The data connection does not exist.                            | 数据连接不存在。         | 请检查数据连接。          |
| 400 | DLF.6323 | The connection type cannot be modified.                        | 连接类型不允许修改。       | 请检查连接类型。          |

| 状态码 | 错误码       | 错误信息                                                                                                 | 描述                                | 处理措施                     |
|-----|-----------|------------------------------------------------------------------------------------------------------|-----------------------------------|--------------------------|
| 400 | DLF.6416  | The cluster has been occupied by another connection.                                                 | 集群已经被其他连接占用了。                     | 请检查集群。                   |
| 400 | DLF.6418  | Either the DWS cluster name or the access address and port number of the cluster must be configured. | 配置DWS集群名或者集群的访问地址、端口，两者二选一，不能都为空。 | 请检查集群DWS集群名或者集群的访问地址、端口。 |
| 400 | DLF.6423  | An exception occurs when the agent is creating a job.                                                | 创建Agent作业异常。                      | 请稍后重试，或者检查Agent状态是否正常。   |
| 400 | DLF.6888  | Invalid account permissions.                                                                         | 账户权限不合法。                          | 请检查账户权限。                 |
| 400 | DLF.21123 | The single node job type cannot be empty.                                                            | 单节点作业类型不能为空。                      | 选择单节点作业类型。               |