

开天集成工作台

## API 参考

文档版本 06  
发布日期 2024-04-24



版权所有 © 华为云计算技术有限公司 2024。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。

# 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

# 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 终端节点	1
1.4 约束与限制	1
1.5 基本概念	1
<b>2 API 概览</b>	<b>3</b>
<b>3 如何调用 API</b>	<b>4</b>
3.1 构造请求	4
3.2 认证鉴权	7
3.3 返回结果	8
<b>4 API</b>	<b>10</b>
4.1 流管理	10
4.1.1 查询流运行历史记录	10
4.1.2 查询单个流运行历史记录	14
4.1.3 获取流运行详细信息	17
4.1.4 提交运行流	23
4.1.5 开启流	25
4.1.6 关闭流	27
4.1.7 webhook 触发流调用（废弃）	29
4.2 应用模型	31
4.2.1 实体管理	31
4.2.1.1 添加实体	31
4.2.1.2 更新实体	36
4.2.1.3 删除实体	42
4.2.1.4 批量删除实体	45
4.2.2 关系管理	47
4.2.2.1 关系操作接口	47
4.2.3 特征管理	50
4.2.3.1 特征操作接口	50
<b>5 应用示例</b>	<b>56</b>
5.1 运行流并查看运行记录	56

<b>6 权限策略和授权项</b> .....	<b>59</b>
6.1 权限及授权项说明.....	59
6.2 流管理权限.....	60
<b>7 附录</b> .....	<b>61</b>
7.1 状态码.....	61
7.2 错误码.....	63
7.3 获取项目 ID.....	67
7.4 获取账号 ID.....	68
<b>8 历史 API</b> .....	<b>69</b>
8.1 查询流历史活动记录.....	69
8.2 查询单个流历史活动记录.....	72
8.3 获取流运行详细信息.....	75
<b>9 修订记录</b> .....	<b>80</b>

# 1 使用前必读

## 1.1 概述

欢迎使用开天集成工作台（MacroVerse SmartStage for Integrators, MSSI），开天集成工作台是一个助力企业应用创新的开放的集成平台。基于元数据、可配置的集成框架，汇聚丰富的集成资产以及开发工具，帮助企业实现业务流程的自动化，连通企业内的应用孤岛，以及创新应用的高效（低代码）开发。

您可以使用本文档提供的API对MSSI服务进行相关操作，如查询、提交、启用流。MSSI服务支持的全部操作请参见[API概览](#)。

在调用开天集成工作台API之前，请确保已经充分了解开天集成工作台相关概念，详细信息请参见《[产品介绍](#)》。

## 1.2 调用说明

开天集成工作台提供了REST（Representational State Transfer）风格API，支持通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

## 1.3 终端节点

终端节点即调用API的请求地址，不同服务不同区域的终端节点不同。MSSI服务目前仅部署在“华北-北京四”区域，Endpoint为“mssi.cn-north-4.myhuaweicloud.com”。

## 1.4 约束与限制

无。

## 1.5 基本概念

- 账号  
用户注册时的账号，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。由于账号是付费主体，为了确保账号安全，建

议不要直接使用账号进行日常工作，而是创建并使用用户进行日常工作。

- IAM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[我的凭证](#)下，可以查看账号ID和IAM用户ID。通常在调用API的鉴权过程中，需要用到账号、用户和密码等信息。

- 区域

指云资源所在的物理位置，同一区域内可用区间内网互通，不同区域间内网不互通。通过在不同地区创建云资源，可以将应用程序设计的更接近特定客户的要求，或满足不同地区的法律或其他要求。

- 可用区

可用区是同一服务区内，电力和网络互相独立的地理区域，一般是一个独立的物理机房，这样可以保证可用区的独立性。

一个区域内有多个可用区，一个可用区发生故障后不会影响同一区域内下的其它可用区。

可用区间通过内网访问。

- 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，用户可以访问账号中该区域的所有资源。如果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，并在子项目中购买资源，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。

同样在[我的凭证](#)下，可以查看项目ID。

- 企业项目

企业项目是项目的升级版，针对企业不同项目间资源的分组和管理，是逻辑隔离。企业项目中可以包含多个区域的资源，且项目中的资源可以迁入迁出。

关于企业项目ID的获取及企业项目特性的详细信息，请参见《[企业管理服务用户指南](#)》。

# 2 API 概览

开天集成工作台接口的分类与说明如表2-1所示。

表 2-1 API 概览

类型	子类型	说明
流管理	流管理	包含查询流活动记录、查询流运行信息、运行流、禁用流等。
应用模型	实体管理	包含添加实体、更新实体、删除实体等接口。
	关系管理	包含关系操作接口。
	特征管理	包含特征操作接口。

# 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**获取用户Token**接口说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

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 请求 URI

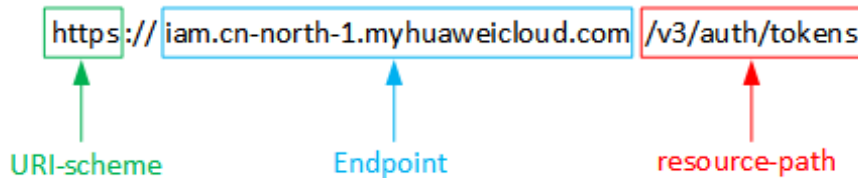
参数	说明
URI-scheme	表示用于传输请求的协议，当前所有API均采用 <b>HTTPS</b> 协议。
Endpoint	指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从 <b>地区和终端节点</b> 中获取。例如IAM服务在“华北-北京一”区域的Endpoint为“iam.cn-north-1.myhuaweicloud.com”。
resource-path	资源路径，也即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。
query-string	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。



例如您需要获取IAM在“华北-北京一”区域的Token，则需使用“华北-北京一”区域的Endpoint（iam.cn-north-1.myhuaweicloud.com），并在[获取用户Token](#)的URI部分找到resource-path（/v3/auth/tokens），拼接起来如下所示。

```
https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

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），它告诉服务您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[获取用户Token](#)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-1.myhuaweicloud.com/v3/auth/tokens
```

## 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。

需要添加到请求中的公共消息头如[表3-3](#)所示。

表 3-3 公共请求消息头

参数名	说明	是否必选	示例
Content-type	消息体的类型（格式），默认取值为“application/json”。	是	application/json
X-Auth-Token	用户Token，也就是调用 <b>获取用户Token</b> 接口的响应值，该接口是唯一不需要认证的接口。	否，使用Token认证时该字段必选。	-
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。	否，如果使用AK/SK做接口认证的时候，那么此字段必须设置；如果使用PKI token的时候，不必设置。	20190307T101459Z
Authorization	签名认证信息。 该值来源于请求签名结果，使用AK/SK进行加密签名的时候需要。 类型：字符串 默认值：无	否，使用AK/SK认证时必选。	SDK-HMAC-SHA256 Credential=ZIRRKMT WPTQFQ11WKNKB/ 20150907//ec2/ sdk_request, SignedHeaders=content-type;host;x-sdk-date, Signature=55741b610f 3c9fa3ae40b5a8021eb f7ebc2a28a603fc62d2 5cb3bfe6608e1994
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。	否，使用AK/SK认证时必选。	code.test.com 或者 code.test.com:443

### 📖 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[API签名指南](#)。

对于**获取用户Token**接口，由于不需要认证，所以只添加“Content-Type”即可，添加消息头后的请求如下所示。

```
POST https://mssi.cn-north-4.myhuaweicloud.com/v1/{project_id}/flows/{flow_id}/run
Content-Type: application/json
```

## 请求消息体

请求消息体通常以结构化格式发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。如果请求消息体中参数支持中文，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[获取用户Token](#)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，其中***username***为用户名，***domainname***为用户所属的账号名称，***\*\*\*\*\****为用户登录密码，***xxxxxxxx***为project的名称，如cn-north-1，您可以从[地区和终端节点](#)中获取。

### 📖 说明

scope参数定义了Token的作用域，下面示例中获取的Token仅能访问project下的资源。您还可以设置Token作用域为某个账号下所有资源或账号的某个project下的资源，详细定义请参见[获取用户Token](#)。

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type:application/json
```

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

到这里为止这个请求需要的内容就具备齐全了，您可以使用[curl](#)、[Postman](#)或直接编写代码等方式发送请求调用API。对于[获取用户Token](#)接口，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

## 3.2 认证鉴权

调用接口有如下两种认证方式，可以选择其中一种进行认证鉴权。推荐使用AK/SK认证，其安全性比Token认证要高。

- 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。

## Token 认证

### 📖 说明

Token的有效期为24小时，需要使用同一个Token鉴权时，可以缓存起来，避免频繁调用。

获取Token后，再调用其他接口时，需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFGH...”，则调用接口时将“X-Auth-Token: ABCDEFGH...”加到请求消息头即可，如下所示。

```
GET https://mssi.cn-north-4.myhuaweicloud.com/v1/{project_id}/flows/running-history
Content-Type: application/json
X-Auth-Token: ABCDEFGH...
```

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>。

还可以参考华为云IAM认证服务了解如何[获取IAM用户Token](#)。

## AK/SK 认证

### 📖 说明

AK/SK签名认证方式仅支持消息体大小12M以内，12M以上的请求请使用Token认证。

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。

使用AK/SK认证时，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](#)。

## 3.3 返回结果

请求发送以后，会收到响应，包含：状态码、响应消息头和响应消息体。

### 状态码

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[获取用户Token](#)接口，返回如[图3-2](#)所示的消息头，其中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

图 3-2 获取用户 Token 响应消息头

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token → MIIYXQYJKoZIhvcNAQcCoIIYtjCCGEoCAQExDTALBgIghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6IjpwMTktMDItMTNUMC
fj3Kjs6YgKnpVNRbW2eZ5eb78SZ0kqjACgkqO1wi4JlGzrpd18LGXK5txidf4lqHCYb8P4NaY0NYejcAgzIVefYtLWT1GS00zxKZmlQHQ82HBqHdglZO9fuEbL5dMhdavj+33wEI
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jggfFkNPQuFSOU8+uSsttVwRtNfsC+qTp2Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUvhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

## 响应消息体

响应消息体通常以结构化格式返回，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**获取用户Token**接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "cn-north-1",
            .....

```

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": "The format of message is error",
  "error_code": "AS.0001"
}
```

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

# 4 API

## 4.1 流管理

### 4.1.1 查询流运行历史记录

#### 功能介绍

查询租户下的所有流运行的历史记录。

#### 接口约束

无

#### URI

GET /v2/{project\_id}/flows/running-history

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，用于资源隔离。获取方法请参考 <a href="#">获取项目ID</a> 。

表 4-2 Query 参数

参数	是否必选	参数类型	描述
time_range	否	String	时间筛选范围，查询该范围内的流运行记录。UTC时间格式： yyyy-MM-dd'T'HH:mm:ss.SSS'Z'toyyyy-MM-dd'T'HH:mm:ss.SSS'Z'如： 2020-11-10T00:00:00Zto2020-11-12T00:00:00。
name	否	String	流的名称。
status	否	String	流运行状态，枚举值。 <ul style="list-style-type: none"> <li>• success: 流运行成功</li> <li>• fail: 流运行失败</li> </ul>
offset	否	Integer	当前页，起始页数应大于等于0。 最小值：1
limit	否	Integer	每页记录数，每页记录数应大于等于1。 最小值：1

## 请求参数

无

## 响应参数

状态码： 200

表 4-3 响应 Body 参数

参数	参数类型	描述
total	Long	历史记录总数。
histories	Array of <b>histories</b> objects	所有历史记录的信息。

表 4-4 histories

参数	参数类型	描述
id	String	流运行记录ID。

参数	参数类型	描述
flow_id	String	流ID。
flow_name	String	流名称。
flow_version_id	String	流运行版本ID。
run_model	String	流运行模式，通常有手动、自动、测试这几种。手动方式需要用户在页面点击运行流，自动方式会按照触发器的类型自动运行流，测试方式用户在创建流的时候可以测试流的功能。
status	String	最后一次运行，流运行状态。 <ul style="list-style-type: none"> <li>• success: 流运行成功</li> <li>• fail: 流运行失败</li> </ul>
icon	String	流图标。
start_time	String	开始时间。
end_time	String	结束时间。
exec_time	Long	执行时间，单位毫秒。
running_times	Long	运行次数。
last_time	String	最后一次运行时间。
message	String	最后一次运行，输出的错误信息。

状态码： 403

表 4-5 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

状态码： 500



表 4-6 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度：8 最大长度：36
error_msg	String	错误描述。 最小长度：2 最大长度：512

## 请求示例

无

## 响应示例

状态码：200

请求成功。

```
{
  "count" : 5,
  "histories" : [ {
    "id" : "f4ce35fd-5e33-4d9e-b7e1-63f3a07d74ce",
    "flow_id" : "55e42ede-25dd-4f79-9aa4-842e1ba435a1",
    "flow_name" : "业务对象测试",
    "flow_version_id" : null,
    "run_model" : null,
    "status" : "fail",
    "icon" : "data:image/png;base64,iVBORw0KGgoAAA...",
    "start_time" : "2022-02-28T03:34:04.025Z",
    "end_time" : null,
    "exec_time" : 0,
    "running_times" : 1438,
    "last_time" : "2022-02-28T03:34:04.025Z",
    "message" : "trigger action running fail."
  }, "... ]
}
```

## 状态码

状态码	描述
200	请求成功。
403	权限不足。
500	系统异常。

## 错误码

请参见[错误码](#)。

## 4.1.2 查询单个流运行历史记录

### 功能介绍

根据单个流id查询该流的所有运行历史记录。

### 接口约束

无

### URI

GET /v2/{project\_id}/flows/{flow\_id}/running-history

表 4-7 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id, 用于资源隔离。获取方法请参考 <a href="#">获取项目ID</a> 。
flow_id	是	String	流id。

表 4-8 Query 参数

参数	是否必选	参数类型	描述
time_range	否	String	时间筛选范围, 查询该范围内的流运行记录。UTC时间格式: yyyy-MM-dd'T'HH:mm:ss.SSS'Z'toyyyy-MM-dd'T'HH:mm:ss.SSS'Z'如: 2020-11-10T00:00:00Zto2020-11-12T00:00:00。
status	否	String	流运行状态, 枚举值。 <ul style="list-style-type: none"> <li>• success: 流运行成功</li> <li>• fail: 流运行失败</li> </ul>
offset	否	Integer	当前页, 起始页数应大于等于0。 最小值: 1
limit	否	Integer	每页记录数, 每页记录数应大于等于1。 最小值: 1

### 请求参数

无

## 响应参数

状态码： 200

表 4-9 响应 Body 参数

参数	参数类型	描述
total	Long	历史记录总数。
histories	Array of <a href="#">histories</a> objects	所有历史记录的信息。

表 4-10 histories

参数	参数类型	描述
id	String	流运行记录ID。
flow_id	String	流ID。
flow_name	String	流名称。
flow_version_id	String	流运行版本ID。
run_model	String	流运行模式，通常有手动、自动、测试这几种。手动方式需要用户在页面点击运行流，自动方式会按照触发器的类型自动运行流，测试方式用户在创建流的时候可以测试流的功能。
status	String	最后一次运行，流运行状态。 <ul style="list-style-type: none"><li>• success: 流运行成功</li><li>• fail: 流运行失败</li></ul>
icon	String	流图标。
start_time	String	开始时间。
end_time	String	结束时间。
exec_time	Long	执行时间，单位毫秒。
running_times	Long	运行次数。
last_time	String	最后一次运行时间。
message	String	最后一次运行，输出的错误信息。

状态码： 403

表 4-11 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度：8 最大长度：36
error_msg	String	错误描述。 最小长度：2 最大长度：512

状态码：500

表 4-12 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度：8 最大长度：36
error_msg	String	错误描述。 最小长度：2 最大长度：512

## 请求示例

无

## 响应示例

状态码：200

请求成功。

```
{
  "count": 1438,
  "histories": [ {
    "id": "df00b54a-7765-4ed9-98fe-1504119841e4",
    "flow_id": "55e42ede-25dd-4f79-9aa4-842e1ba435a1",
    "flow_name": "业务对象测试",
    "flow_version_id": "55e42ede-25dd-4f79-9aa4-842e1ba435a1",
    "run_model": "auto",
    "status": "fail",
    "icon": null,
    "start_time": "2022-02-28T06:24:04.022Z",
    "end_time": "2022-02-28T06:24:04.022Z",
    "exec_time": 0,
    "running_times": null,
    "last_time": null,
    "message": "trigger action running fail."
  }
  ]
}
```

```
    ]]  
  }
```

## 状态码

状态码	描述
200	请求成功。
403	权限不足。
500	系统异常。

## 错误码

请参见[错误码](#)。

## 4.1.3 获取流运行详细信息

### 功能介绍

根据流id和运行记录id查询流运行详细信息，包括流的Id、名称、图标、开始时间、结束时间、最后一步运行状态以及流步骤元数据。

### 接口约束

无

### URI

GET /v2/{project\_id}/flows/{flow\_id}/running-history/{running\_id}

表 4-13 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id，用于资源隔离。获取方法请参考 <a href="#">获取项目ID</a> 。
flow_id	是	String	流id。
running_id	是	String	流运行记录id。

### 请求参数

无

### 响应参数

状态码： 200

表 4-14 响应 Body 参数

参数	参数类型	描述
id	String	流运行记录id。
flow_id	String	流id。
flow_name	String	流名称。
run_model	String	流运行模式，通常有手动、自动、测试这几种。手动方式需要用户在页面点击运行流，自动方式会按照触发器的类型自动运行流，测试方式用户在创建流的时候可以测试流的功能。
status	String	最后一次运行，流运行状态。 <ul style="list-style-type: none"> <li>• success: 流运行成功</li> <li>• fail: 流运行失败</li> </ul>
icon	String	流图标。
start_time	String	开始运行时间。
end_time	String	结束运行时间。
exec_time	Long	执行时间，单位毫秒。
message	String	最后一次运行，输出的错误信息。
steps	Array of <b>steps</b> objects	一个流通常由多个步骤组成，该数组是流每一步骤的元数据。

表 4-15 steps

参数	参数类型	描述
stepId	String	操作步骤id。
name	String	连接器名字。
actionName	String	操作步骤名称。
actionType	String	操作步骤类型。
call	<b>call</b> object	连接器操作信息。
icon	String	步骤的图标。
transmit	Array of strings	该步骤执行完之后，下一步操作的ID列表。
mockOutput	String	模拟输出数据。

参数	参数类型	描述
lastRunningStatus	String	流中该步骤的最后一次运行状态。 <ul style="list-style-type: none"> <li>• success: 流运行成功</li> <li>• fail: 流运行失败</li> </ul>
cost	Long	执行此步骤的花费时间，单位毫秒。

表 4-16 call

参数	参数类型	描述
connectorRef	String	连接器操作id。
triggerRef	<b>triggerRef</b> object	触发器操作定义。
actionRef	<b>actionRef</b> object	普通操作定义。
connectorVersion	String	连接器版本。

表 4-17 triggerRef

参数	参数类型	描述
refName	String	触发器操作名称。
refId	String	触发器操作id。
type	String	触发器操作类型。
authentication	<b>authentication</b> object	连接器操作认证。
input	Map<String, Object>	连接器执行时的输入数据。
config	Map<String, Object>	连接器配置信息。
customFunctionIds	Array of strings	用户可以在流中使用自定义函数处理数据，该字段存储所有用户在该流中使用的自定义的函数id。
customFunctionMap	Map<String, Object>	自定义函数配置。

表 4-18 authentication

参数	参数类型	描述
connectionId	String	连接器id。
schema	<b>schema</b> object	连接器认证元数据。

表 4-19 schema

参数	参数类型	描述
id	String	连接id，流使用连接器时需要配置认证信息，称为连接。
authInfo	Map<String, Object>	连接认证信息。
connectorId	String	连接器id。
type	String	连接器类型。

表 4-20 actionRef

参数	参数类型	描述
refName	String	普通操作名称。
refId	String	普通操作id。
type	String	普通操作类型。
authentication	<b>authentication</b> object	连接器操作认证。
input	Map<String, Object>	连接器执行时的输入数据。
config	Map<String, Object>	连接器配置信息。
customFunctionIds	Array of strings	用户可以在流中使用自定义函数处理数据，该字段存储所有用户在该流中使用的自定义的函数id。
customFunctionMap	Map<String, Object>	自定义函数配置。



表 4-21 authentication

参数	参数类型	描述
connectionId	String	连接器id。
schema	<b>schema</b> object	连接器认证元数据。

表 4-22 schema

参数	参数类型	描述
id	String	连接id。
authInfo	Map<String, Object>	连接认证信息。
connectorId	String	连接器id。
type	String	连接器类型。

状态码： 403

表 4-23 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

状态码： 500

表 4-24 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36

参数	参数类型	描述
error_msg	String	错误描述。 最小长度：2 最大长度：512

## 请求示例

无

## 响应示例

状态码：200

请求成功。

```
{
  "id": "d96ba061-3caf-49cb-8950-4dac97a8f082",
  "flow_id": "cfdcacae-a44f-4827-8ad9-db47ad2a8b41",
  "flow_name": "遍历集合元素测试",
  "run_model": "manual",
  "status": "success",
  "icon": "https://connector-icon.obs.cn-north-7.ulanhqab.huawei.com/icon/c9710579264845c6a5339fd738e66fd6/155e6930bdb24d9d95b5d3c636e133eb.png",
  "start_time": 1655881122470,
  "end_time": 1655881126839,
  "exec_time": 4369,
  "message": null,
  "steps": [ {
    "stepId": "05778538-83a6-467b-bb92-738d9049bb3c",
    "actionName": "手动触发器触发事件",
    "actionType": null,
    "call": {
      "connectorRef": "ManualTrigger",
      "triggerRef": {
        "refName": "手动触发器触发事件",
        "refId": "ManualTrigger",
        "type": "instant",
        "authentication": {
          "connectionId": null,
          "schema": null
        }
      },
      "input": {
        "body": { }
      }
    },
    "config": null,
    "customFunctionMap": null
  },
  "actionRef": null,
  "connectorVersion": "1"
},
  "transmit": [ "bc744b98-007f-496e-bd10-ed460548f7de" ],
  "icon": "https://connector-icon.obs.cn-north-7.ulanhqab.huawei.com/icon/c9710579264845c6a5339fd738e66fd6/155e6930bdb24d9d95b5d3c636e133eb.png",
  "name": "手动触发流",
  "lastRunningStatus": "success",
  "cost": 354
} ] }
```

## 状态码

状态码	描述
200	请求成功。
403	权限不足。
500	系统异常。

## 错误码

请参见[错误码](#)。

### 4.1.4 提交运行流

#### 功能介绍

根据流id提交流到后台运行。

#### 接口约束

无

#### URI

POST /v1/{project\_id}/flows/{flow\_id}/run

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，用于资源隔离。获取方法请参考 <a href="#">获取项目ID</a> 。
flow_id	是	String	流id。

#### 请求参数

表 4-26 请求 Body 参数

参数	是否必选	参数类型	描述
run_mode	是	String	流运行模式，通常有手动、自动、测试这几种。手动方式需要用户在页面点击运行流，自动方式会按照触发器的类型自动运行流，测试方式用户在创建流的时候可以测试流的功能。

参数	是否必选	参数类型	描述
input	否	Map<String,Object>	流操作输入参数，一般为JSON字符串，可以为空。具体与要执行的流有关。

## 响应参数

状态码： 200

表 4-27 响应 Body 参数

参数	参数类型	描述
running_id	String	流记录id。
flow_id	String	流id。
success	Boolean	流运行状态 <ul style="list-style-type: none"><li>• success: 流运行成功</li><li>• fail: 流运行失败</li></ul>
flow_name	String	流名称。
message	String	返回信息。

状态码： 403

表 4-28 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

状态码： 500

表 4-29 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度：8 最大长度：36
error_msg	String	错误描述。 最小长度：2 最大长度：512

## 请求示例

无

## 响应示例

状态码：200

请求成功。

```
{  
  "running_id" : "0d16aee0-bdb9-4a1c-851b-87e11ec67322",  
  "flow_id" : "f113423c-4759-4fe3-95fb-642bb265a2d2",  
  "success" : true,  
  "flow_name" : "流9472f",  
  "message" : ""  
}
```

## 状态码

状态码	描述
200	请求成功。
403	权限不足。
500	系统异常。

## 错误码

请参见[错误码](#)。

## 4.1.5 开启流

### 功能介绍

根据流id开启流，开启的流才可以运行。

## 接口约束

无

## URI

POST /v1/{project\_id}/flows/{flow\_id}/enable

表 4-30 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id, 用于资源隔离。获取方法请参考 <a href="#">获取项目ID</a> 。
flow_id	是	String	流id。

## 请求参数

无

## 响应参数

状态码： 200

表 4-31 响应 Body 参数

参数	参数类型	描述
success	Boolean	响应是否成功, 默认为true。

状态码： 403

表 4-32 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

状态码： 500

表 4-33 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度：8 最大长度：36
error_msg	String	错误描述。 最小长度：2 最大长度：512

## 请求示例

无

## 响应示例

状态码：200

请求成功。

```
{  
  "success" : true,  
  "message" : null  
}
```

## 状态码

状态码	描述
200	请求成功。
403	权限不足。
500	系统异常。

## 错误码

请参见[错误码](#)。

## 4.1.6 关闭流

### 功能介绍

根据流id关闭流，关闭的流不能运行。

### 接口约束

无

## URI

POST /v1/{project\_id}/flows/{flow\_id}/disable

表 4-34 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id, 用于资源隔离。获取方法请参考 <a href="#">获取项目ID</a> 。
flow_id	是	String	流id。

## 请求参数

无

## 响应参数

状态码: 200

表 4-35 响应 Body 参数

参数	参数类型	描述
success	Boolean	响应是否成功, 默认为true。

## 请求示例

无

## 响应示例

状态码: 200

请求成功。

```
{  
  "success" : true,  
  "message" : null  
}
```

## 状态码

状态码	描述
200	请求成功。



## 错误码

请参见[错误码](#)。

## 4.1.7 webhook 触发流调用（废弃）

### 功能介绍

webhook触发流调用，该接口在集成工作台新版本中已经废弃。

### 接口约束

无

### URI

POST /v1/{project\_id}/flows/webhook/trigger/{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，用于资源隔离。获取方法请参考 <a href="#">获取项目ID</a> 。
id	是	String	webhook id。

### 请求参数

表 4-37 请求 Body 参数

参数	是否必选	参数类型	描述
input	否	String	api流入参。

### 响应参数

状态码： 200

表 4-38 响应 Body 参数

参数	参数类型	描述
success	Boolean	响应是否成功，默认为true。

状态码： 403

表 4-39 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度：8 最大长度：36
error_msg	String	错误描述。 最小长度：2 最大长度：512

状态码：500

表 4-40 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度：8 最大长度：36
error_msg	String	错误描述。 最小长度：2 最大长度：512

## 请求示例

无

## 响应示例

状态码：200

请求成功。

```
{  
  "success" : true,  
  "message" : null  
}
```

## 状态码

状态码	描述
200	请求成功。
403	权限不足。
500	系统异常。

## 错误码

请参见[错误码](#)。

## 4.2 应用模型

### 4.2.1 实体管理

#### 4.2.1.1 添加实体

#### 功能介绍

添加实体。

#### URI

POST /v3/{project\_id}/abm/instances/{instance\_id}/metadata3/entities

表 4-41 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参考 <a href="#">获取项目ID</a> 。
instance_id	是	String	实例的ID。 <ul style="list-style-type: none"><li>• 最小长度：1</li><li>• 最大长度：64</li></ul>

#### 请求参数

表 4-42 请求 Body 参数

参数	是否必选	参数类型	描述
entities	是	Array of <a href="#">表 4-43</a> objects	实体请求列表，最大个数不超过200个。

表 4-43 MetadataEntityRequest

参数	是否必选	参数类型	描述
package_id	是	String	模型包ID。 <ul style="list-style-type: none"> <li>最小长度: 1</li> <li>最大长度: 50</li> </ul>
entity_name	是	String	实体名称。 <ul style="list-style-type: none"> <li>最小长度: 1</li> <li>最大长度: 200</li> </ul>
entity_type	是	String	实体类型。 <ul style="list-style-type: none"> <li>最小长度: 1</li> <li>最大长度: 150</li> </ul>
subtype	否	String	实体子类型。 最大长度: 50
entity_attributes	否	Array of <a href="#">表 4-45</a> objects	实体属性。
signatures	否	Array of <a href="#">表 4-44</a> objects	特征。

表 4-44 MetadataSignature

参数	是否必选	参数类型	描述
signature_type	是	String	特征类型。
signature_name	否	String	特征名称。
signature_attributes	否	Array of <a href="#">表 4-45</a> objects	特征属性。

表 4-45 MetadataAttributeRequest

参数	是否必选	参数类型	描述
attribute_id	否	String	属性ID。 最大长度: 200
attribute_value	否	String	属性值。

## 响应参数

状态码： 200

表 4-46 响应 Body 参数

参数	参数类型	描述
success	Array of 表 4-47 objects	操作成功列表。
failure	Array of 表 4-48 objects	操作失败列表。

表 4-47 MetadataEntitySimpleResponse

参数	参数类型	描述
entity_urn	String	实体urn。
entity_name	String	实体名称。
entity_type	String	实体类别。

表 4-48 MetadataEntityFailureResp

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。
entity_name	String	实体名称。
entity_type	String	实体类别。

状态码： 400

表 4-49 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： 500

表 4-50 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。

## 请求示例

POST https://{endpoint}/v3/{project\_id}/abm/instances/{instance\_id}/metadata3/entities

```
{
  "entities": [
    {
      "package_id": "Metamodel3",
      "entity_name": "q8",
      "entity_type": "Metamodel3-SubjectArea",
      "entity_attributes": [
        {
          "attribute_id": "Metamodel3-SubjectArea-NAME",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-SubjectArea-PermissionCode",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-SubjectArea-OrderNumber",
          "attribute_value": ""
        },
        {
          "attribute_id": "entity_name",
          "attribute_value": "q8"
        }
      ]
    },
    {
      "signature_name": "",
      "signature_type": "Metamodel3-descriptionInfo",
      "signature_attributes": [
        {
          "attribute_id": "Metamodel3-descriptionInfo-NAME",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-descriptionInfo-description",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-descriptionInfo-nameEn",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-descriptionInfo-nameCn",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-descriptionInfo-dataAssetNumber",
          "attribute_value": ""
        }
      ]
    }
  ],
  "signature_name": "",
  "signature_type": "Metamodel3-maintenanceInfo",
  "signature_attributes": [
```

```
{
  {
    "attribute_id": "Metamodel3-maintenanceInfo-NAME",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-createdBy",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-createTime",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updatedBy",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updateTime",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-creatorAccount",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updaterAccount",
    "attribute_value": ""
  }
}
]
```

## 响应示例

状态码: 200

successful operation

```
{
  "msg": null,
  "success": [
    {
      "entity_type": "Metamodel3-SubjectArea",
      "entity_urn": "urn:abm:Metamodel3-SubjectArea:b2570da68c011fbf7bf8e784e9cd2c",
      "entity_name": "q8"
    }
  ],
  "failure": []
}
```

## 状态码

状态码	描述
200	successful operation.
400	Bad Request.
500	Internal Server Error.

## 错误码

请参见[错误码](#)。

### 4.2.1.2 更新实体

#### 功能介绍

更新实体。

#### URI

PUT /v3/{project\_id}/abm/instances/{instance\_id}/metadata3/entities/{entity\_urn}

表 4-51 路径参数

参数	是否必选	参数类型	描述
entity_urn	是	String	实例urn。
project_id	是	String	项目ID，获取方法请参考 <a href="#">获取项目ID</a> 。
instance_id	是	String	实例的ID。 <ul style="list-style-type: none"><li>• 最小长度：1</li><li>• 最大长度：64</li></ul>

#### 请求参数

表 4-52 请求 Body 参数

参数	是否必选	参数类型	描述
package_id	是	String	模型包ID。 <ul style="list-style-type: none"><li>• 最小长度：1</li><li>• 最大长度：50</li></ul>
entity_name	是	String	实体名称。 <ul style="list-style-type: none"><li>• 最小长度：1</li><li>• 最大长度：200</li></ul>
entity_type	是	String	实体类型。 <ul style="list-style-type: none"><li>• 最小长度：1</li><li>• 最大长度：150</li></ul>
subtype	否	String	实体子类型。 最大长度：50



参数	是否必选	参数类型	描述
entity_attributes	否	Array of <a href="#">表 4-54</a> objects	实体属性。
signatures	否	Array of <a href="#">表 4-53</a> objects	特征。

表 4-53 MetadataSignature

参数	是否必选	参数类型	描述
signature_type	是	String	特征类型。 最大长度：150
signature_name	否	String	特征名称。 最大长度：200
signature_attributes	否	Array of <a href="#">表 4-54</a> objects	特征属性。

表 4-54 MetadataAttributeRequest

参数	是否必选	参数类型	描述
attribute_id	否	String	属性ID。 最大长度：200
attribute_value	否	String	属性值。

## 响应参数

状态码： 200

表 4-55 响应 Body 参数

参数	参数类型	描述
package_id	String	模型包ID。
entity_urn	String	实体urn。
entity_name	String	实体名称。
entity_type	String	实体类别。
entity_attributes	Array of <a href="#">表 4-57</a> objects	实体属性。

参数	参数类型	描述
signatures	Array of <a href="#">表 4-56</a> objects	特征。

表 4-56 MetadataEntitySignatureResp

参数	参数类型	描述
signature_name	String	特征名称。
signature_type	String	特征类型。
signature_attributes	Array of <a href="#">表 4-57</a> objects	特征属性。

表 4-57 MetadataAttributeResp

参数	参数类型	描述
attribute_id	String	属性ID。
attribute_value	String	属性值。
attribute_code	String	属性编码。
attribute_name	String	属性名称。
attribute_name_en	String	属性英文名称。

状态码： 400

表 4-58 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： 500

表 4-59 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。

## 请求示例

PUT https://{endpoint}/v3/{project\_id}/abm/instances/{instance\_id}/metadata3/entities/{entity\_urn}

```
{
  "package_id": "Metamodel3",
  "entity_name": "主题域02",
  "entity_type": "Metamodel3-SubjectArea",
  "entity_attributes": [
    {
      "attribute_id": "Metamodel3-SubjectArea-NAME",
      "attribute_value": ""
    },
    {
      "attribute_id": "Metamodel3-SubjectArea-PermissionCode",
      "attribute_value": ""
    },
    {
      "attribute_id": "Metamodel3-SubjectArea-OrderNumber",
      "attribute_value": ""
    },
    {
      "attribute_id": "entity_name",
      "attribute_value": "主题域02"
    }
  ],
  "signatures": [
    {
      "signature_name": "测试",
      "signature_type": "Metamodel3-descriptionInfo",
      "signature_attributes": [
        {
          "attribute_id": "Metamodel3-descriptionInfo-NAME",
          "attribute_value": "测试"
        },
        {
          "attribute_id": "Metamodel3-descriptionInfo-description",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-descriptionInfo-nameEn",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-descriptionInfo-nameCn",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-descriptionInfo-dataAssetNumber",
          "attribute_value": ""
        }
      ]
    }
  ],
  {
    "signature_name": "",
    "signature_type": "Metamodel3-maintenanceInfo",
    "signature_attributes": [
      {
        "attribute_id": "Metamodel3-maintenanceInfo-NAME",
```

```
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-createdBy",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-createTime",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updatedBy",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updateTime",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-creatorAccount",
    "attribute_value": ""
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updaterAccount",
    "attribute_value": ""
  }
]
}
```

## 响应示例

**状态码： 200**

successful operation

```
{
  "entity_type": "Metamodel3-SubjectArea",
  "entity_attributes": [
    {
      "attribute_id": "Metamodel3-SubjectArea-NAME",
      "attribute_name_en": "NAME",
      "attribute_name": "名称",
      "attribute_value": "",
      "attribute_code": "NAME"
    },
    {
      "attribute_id": "Metamodel3-SubjectArea-PermissionCode",
      "attribute_name_en": "PermissionCode",
      "attribute_name": "权限编码",
      "attribute_value": "",
      "attribute_code": "PermissionCode"
    },
    {
      "attribute_id": "Metamodel3-SubjectArea-OrderNumber",
      "attribute_name_en": "OrderNumber",
      "attribute_name": "顺序值",
      "attribute_value": "",
      "attribute_code": "OrderNumber"
    },
    {
      "attribute_id": "entity_name",
      "attribute_name_en": null,
      "attribute_name": null,
      "attribute_value": "主题域02",
      "attribute_code": null
    }
  ]
},
```

```
"entity_urn": "urn:abm:Metamodel3-SubjectArea:e0ee33d38c01167db2c63695f20929",
"entity_name": "主题域02",
"package_id": "Metamodel3",
"signatures": [
  {
    "signature_attributes": [
      {
        "attribute_id": "Metamodel3-descriptionInfo-NAME",
        "attribute_name_en": "NAME",
        "attribute_name": "名称",
        "attribute_value": "测试",
        "attribute_code": "NAME"
      },
      {
        "attribute_id": "Metamodel3-descriptionInfo-description",
        "attribute_name_en": "Description",
        "attribute_name": "描述内容",
        "attribute_value": "",
        "attribute_code": "description"
      },
      {
        "attribute_id": "Metamodel3-descriptionInfo-nameEn",
        "attribute_name_en": "NameEn",
        "attribute_name": "英文名",
        "attribute_value": "",
        "attribute_code": "nameEn"
      },
      {
        "attribute_id": "Metamodel3-descriptionInfo-nameCn",
        "attribute_name_en": "NameCn",
        "attribute_name": "中文名",
        "attribute_value": "",
        "attribute_code": "nameCn"
      },
      {
        "attribute_id": "Metamodel3-descriptionInfo-dataAssetNumber",
        "attribute_name_en": "DataAssetNumber",
        "attribute_name": "资产编码",
        "attribute_value": "",
        "attribute_code": "dataAssetNumber"
      }
    ],
    "signature_name": "测试",
    "version": "0",
    "signature_type": "Metamodel3-descriptionInfo"
  },
  {
    "signature_attributes": [
      {
        "attribute_id": "Metamodel3-maintenanceInfo-NAME",
        "attribute_name_en": "NAME",
        "attribute_name": "名称",
        "attribute_value": "",
        "attribute_code": "NAME"
      },
      {
        "attribute_id": "Metamodel3-maintenanceInfo-createdBy",
        "attribute_name_en": "CreatedBy",
        "attribute_name": "创建人",
        "attribute_value": "",
        "attribute_code": "createdBy"
      },
      {
        "attribute_id": "Metamodel3-maintenanceInfo-createTime",
        "attribute_name_en": "CreateTime",
        "attribute_name": "创建时间",
        "attribute_value": "",
        "attribute_code": "createTime"
      }
    ],
  }
]
```

```
{
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updatedBy",
    "attribute_name_en": "UpdatedBy",
    "attribute_name": "修改人",
    "attribute_value": "",
    "attribute_code": "updatedBy"
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updateTime",
    "attribute_name_en": "UpdateTime",
    "attribute_name": "修改时间",
    "attribute_value": "",
    "attribute_code": "updateTime"
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-creatorAccount",
    "attribute_name_en": "CreatorAccount",
    "attribute_name": "创建账号",
    "attribute_value": "",
    "attribute_code": "creatorAccount"
  },
  {
    "attribute_id": "Metamodel3-maintenanceInfo-updaterAccount",
    "attribute_name_en": "UpdaterAccount",
    "attribute_name": "修改账号",
    "attribute_value": "",
    "attribute_code": "updaterAccount"
  }
],
"signature_name": "",
"version": "0",
"signature_type": "Metamodel3-maintenanceInfo"
}
]
```

## 状态码

状态码	描述
200	successful operation.
400	Bad Request.
500	Internal Server Error.

## 错误码

请参见[错误码](#)。

### 4.2.1.3 删除实体

## 功能介绍

删除实体。

## URI

DELETE /v3/{project\_id}/abm/instances/{instance\_id}/metadata3/entities/  
{entity\_urn}

表 4-60 路径参数

参数	是否必选	参数类型	描述
entity_urn	是	String	实例urn。
project_id	是	String	项目ID，获取方法请参考 <a href="#">获取项目ID</a> 。
instance_id	是	String	实例的ID。 <ul style="list-style-type: none"> <li>• 最小长度：1</li> <li>• 最大长度：64</li> </ul>

## 请求参数

无

## 响应参数

状态码： 200

表 4-61 响应 Body 参数

参数	参数类型	描述
acknowledged	Boolean	true 执行成功/false 执行失败。
msg	String	操作结果信息。
result	<a href="#">表4-62</a> object	返回结果体。

表 4-62 ResultResp

参数	参数类型	描述
msg	String	提示信息。
success	Array of strings	删除成功的urn列表。
failure	Array of strings	删除失败的urn列表。

状态码： 400

表 4-63 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。

状态码： 500

表 4-64 响应 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

```
DELETE https://{endpoint}/v3/{project_id}/abm/instances/{instance_id}/metadata3/entities/{entity_urn}
```

## 响应示例

状态码： 200

successful operation

```
{
  "msg": "",
  "result": {
    "msg": "",
    "success": [
      "urn:abm:Metamodel3-SubjectArea:f841abee8d019f1a94c772d029db92"
    ],
    "failure": []
  },
  "acknowledged": true
}
```

## 状态码

状态码	描述
200	successful operation.
400	Bad Request.
500	Internal Server Error.

## 错误码

请参见[错误码](#)。



## 4.2.1.4 批量删除实体

### 功能介绍

批量删除实体。

### URI

POST /v3/{project\_id}/abm/instances/{instance\_id}/metadata3/entities/delete

表 4-65 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参考 <a href="#">获取项目ID</a> 。
instance_id	是	String	实例的ID。 <ul style="list-style-type: none"><li>• 最小长度：1</li><li>• 最大长度：64</li></ul>

### 请求参数

表 4-66 请求 Body 参数

参数	是否必选	参数类型	描述
entity_urns	否	Array of strings	实体urn列表。 数组长度：0 - 200

### 响应参数

状态码： 200

表 4-67 响应 Body 参数

参数	参数类型	描述
acknowledged	Boolean	true 执行成功/false 执行失败。
msg	String	操作结果信息。
result	<a href="#">表4-68</a> object	返回结果体。

表 4-68 ResultResp

参数	参数类型	描述
msg	String	提示信息。
success	Array of strings	删除成功的urn列表。
failure	Array of strings	删除失败的urn列表。

状态码： 400

表 4-69 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。

状态码： 500

表 4-70 响应 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

```
POST https://{endpoint}/v3/{project_id}/abm/instances/{instance_id}/metadata3/entities/delete
{
  "entity_urns": [
    "urn:abm:Metamodel3-SubjectArea:1c13a4a68c01c6d82e4ef95e5fac8c",
    "urn:abm:Metamodel3-SubjectArea:b2570da68c011fbf7bf8e784e9cd2c"
  ]
}
```

## 响应示例

状态码： 200

successful operation

```
{
  "msg": "",
  "result": {
    "msg": "",
    "success": [
```

```
"urn:abm:Metamodel3-SubjectArea:1c13a4a68c01c6d82e4ef95e5fac8c",  
"urn:abm:Metamodel3-SubjectArea:b2570da68c011fbf7bf8e784e9cd2c"  
],  
"failure": []  
},  
"acknowledged": true  
}
```

## 状态码

状态码	描述
200	successful operation.
400	Bad Request.
500	Internal Server Error.

## 错误码

请参见[错误码](#)。

## 4.2.2 关系管理

### 4.2.2.1 关系操作接口

#### 功能介绍

关系操作接口。

#### URI

POST /v3/{project\_id}/abm/instances/{instance\_id}/metadata3/relationships

表 4-71 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参考 <a href="#">获取项目ID</a> 。
instance_id	是	String	实例的ID。 <ul style="list-style-type: none"><li>最小长度：1</li><li>最大长度：64</li></ul>

## 请求参数

表 4-72 请求 Body 参数

参数	是否必选	参数类型	描述
relationships	是	Array of <a href="#">表 4-73</a> objects	关系，最大个数不超过200个。

表 4-73 MetadataRelationshipRequest

参数	是否必选	参数类型	描述
from_urn	是	String	起点urn。 <ul style="list-style-type: none"> <li>最小长度：1</li> <li>最大长度：2000</li> </ul>
target_urn	是	String	终点urn。 <ul style="list-style-type: none"> <li>最小长度：1</li> <li>最大长度：2000</li> </ul>
relationship_code	是	String	关系编码。 <ul style="list-style-type: none"> <li>最小长度：1</li> <li>最大长度：255</li> </ul>
relationship_extend	否	String	关系扩展。
change_type	是	String	操作类型。 枚举值： <ul style="list-style-type: none"> <li>UPSERT</li> <li>DELETE</li> </ul>

## 响应参数

状态码： 200

表 4-74 响应 Body 参数

参数	参数类型	描述
success	Array of <a href="#">表 4-75</a> objects	操作成功列表。
failure	Array of <a href="#">表 4-76</a> objects	操作失败列表。
msg	String	提示信息。

表 4-75 MetadataRelationshipResponse

参数	参数类型	描述
from_urn	String	起点urn。
target_urn	String	终点urn。
relationship_code	String	关系编码。
relationship_extend	String	关系扩展。

表 4-76 MetadataRelationshipFailureResponse

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。
from_urn	String	起点urn。
target_urn	String	终点urn。
relationship_code	String	关系编码。

状态码： 400

表 4-77 响应 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 4-78 响应 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

```
POST https://{endpoint}/v3/{project_id}/abm/instances/{instance_id}/metadata3/relationships
{
  "relationships": [
    {
      "from_urn": "urn:abm:Metamodel3-Report:3b0d3d1e8c0143dda08cc8ae9d058e",
      "target_urn": "urn:abm:Metamodel3-ReportGroup:a40d2f1e8c0143dda08cc8aeb401e4",
      "relationship_code": "Metamodel3-Report-Metamodel3-ReportGroup_CO",
      "change_type": "UPSERT"
    }
  ]
}
```

## 响应示例

状态码： 200

successful operation

```
{
  "msg": null,
  "success": [
    {
      "target_urn": "urn:abm:Metamodel3-ReportGroup:a40d2f1e8c0143dda08cc8aeb401e4",
      "relationship_extend": null,
      "from_urn": "urn:abm:Metamodel3-Report:3b0d3d1e8c0143dda08cc8ae9d058e",
      "relationship_code": "Metamodel3-Report-Metamodel3-ReportGroup_CO"
    }
  ],
  "failure": []
}
```

## 状态码

状态码	描述
200	successful operation.
400	Bad Request.
500	Internal Server Error.

## 错误码

请参见[错误码](#)。

## 4.2.3 特征管理

### 4.2.3.1 特征操作接口

## 功能介绍

特征操作接口。

## URI

POST /v3/{project\_id}/abm/instances/{instance\_id}/metadata3/signatures

表 4-79 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参考 <a href="#">获取项目ID</a> 。
instance_id	是	String	实例的ID。 <ul style="list-style-type: none"> <li>• 最小长度：1</li> <li>• 最大长度：64</li> </ul>

## 请求参数

表 4-80 请求 Body 参数

参数	是否必选	参数类型	描述
package_id	是	String	模型包ID。 <ul style="list-style-type: none"> <li>• 最小长度：1</li> <li>• 最大长度：50</li> </ul>
entity_urn	是	String	实体urn。 <ul style="list-style-type: none"> <li>• 最小长度：1</li> <li>• 最大长度：2000</li> </ul>
signatures	否	Array of <a href="#">表 4-81</a> objects	特征。

表 4-81 MetadataSignatureChangeReq

参数	是否必选	参数类型	描述
change_type	否	String	操作类型。 枚举值： <ul style="list-style-type: none"> <li>• UPSERT</li> <li>• DELETE</li> <li>• CREATE</li> <li>• UPDATE</li> <li>• RESTATE</li> </ul>

参数	是否必选	参数类型	描述
signature_type	是	String	特征类型。 <ul style="list-style-type: none"> <li>• 最小长度: 1</li> <li>• 最大长度: 150</li> </ul>
signature_name	否	String	特征名称。
signature_attributes	否	Array of <a href="#">表 4-82</a> objects	特征属性。

表 4-82 MetadataAttributeRequest

参数	是否必选	参数类型	描述
attribute_id	是	String	属性ID。 最大长度: 200
attribute_value	否	String	属性值。

## 响应参数

状态码: 200

表 4-83 响应 Body 参数

参数	参数类型	描述
success	Array of <a href="#">表 4-84</a> objects	操作成功列表。
failure	Array of <a href="#">表 4-86</a> objects	操作失败列表。
msg	String	提示信息。

表 4-84 MetadaSignatureResp

参数	参数类型	描述
signature_name	String	特征名称。
signature_type	String	特征类型。



参数	参数类型	描述
signature_attributes	Array of <a href="#">表 4-85</a> objects	特征属性。

表 4-85 MetadataAttributeResp

参数	参数类型	描述
attribute_id	String	属性ID。
attribute_value	String	属性值。
attribute_code	String	属性编码。
attribute_name	String	属性名称。

表 4-86 MetadataSignatureFailureResp

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。
signature_name	String	特征名称。
signature_type	String	特征类型。

状态码： 400

表 4-87 响应 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 4-88 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

```
POST https://{endpoint}/v3/{project_id}/abm/instances/{instance_id}/metadata3/signatures
{
  "package_id": "Metamodel3",
  "entity_urn": "urn:abm:Metamodel3-Dataset:(urn:abm:Metamodel3-DataPlatformInstance:
(MYSQL,3ca1d906860167e88b9bfb0577a69),def.sys.x$wait_classes_global_by_avg_latency)",
  "entity_type": "Metamodel3-Dataset",
  "signatures": [
    {
      "change_type": "UPSERT",
      "signature_type": "Metamodel3-securityPrivacy",
      "signature_name": "密级隐私",
      "signature_attributes": [
        {
          "attribute_id": "Metamodel3-securityPrivacy-NAME",
          "attribute_name": "名称",
          "attribute_value": "tesk0"
        },
        {
          "attribute_id": "Metamodel3-securityPrivacy-securityLevel",
          "attribute_name": "密级",
          "attribute_value": ""
        },
        {
          "attribute_id": "Metamodel3-securityPrivacy-privacyLevel",
          "attribute_name": "隐私分级",
          "attribute_value": ""
        }
      ]
    }
  ]
}
```

## 响应示例

状态码: 200

successful operation

```
{
  "msg": null,
  "success": [
    {
      "signature_attributes": [
        {
          "attribute_id": "Metamodel3-securityPrivacy-NAME",
          "attribute_name_en": "NAME",
          "attribute_name": "名称",
          "attribute_value": "tesk0",
          "attribute_code": "NAME"
        },
        {
          "attribute_id": "Metamodel3-securityPrivacy-securityLevel",
          "attribute_name_en": "SecurityLevel",
          "attribute_name": "密级",
          "attribute_value": ""
        }
      ]
    }
  ]
}
```

```
    "attribute_code": "securityLevel"
  },
  {
    "attribute_id": "Metamodel3-securityPrivacy-privacyLevel",
    "attribute_name_en": "PrivacyLevel",
    "attribute_name": "隐私分级",
    "attribute_value": "",
    "attribute_code": "privacyLevel"
  }
],
"signature_name": "密级隐私",
"signature_type": "Metamodel3-securityPrivacy"
}
},
"failure": []
}
```

## 状态码

状态码	描述
200	successful operation.
400	Bad Request.
500	Internal Server Error.

## 错误码

请参见[错误码](#)。

# 5 应用示例

## 5.1 运行流并查看运行记录

### 场景描述

本章节指导用户通过API运行一个流，并查询流的运行记录。API的调用方法请参见[如何调用API](#)。

### 约束限制

无。

### 涉及接口

运行流并查看运行历史记录涉及以下接口：

- [步骤1.1](#)
- [步骤2.1](#)

### 操作步骤

**步骤1** 提交一个流并查看响应信息。

1. 运行流。

- 接口相关信息

URI格式：POST /v1/{project\_id}/flows/{flow\_id}/run

参数说明请参见[提交运行流](#)。

- 请求示例

```
POST http://{endpoint}/v1/f00ca7a36e*****6d6c5f8da8/flows/cfdcacae-a44f-4827-8ad9-  
db47ad2a8b41/run  
{  
  "run_mode": "manual",  
  "input": {},  
  "trigger_timing": "editFlow"  
}
```

{endpoint}为“mssi.cn-north-4.myhuaweicloud.com”。

- 响应示例

```
{
  "running_id": "d96ba061-3caf-49cb-8950-4dac97a8f082",
  "flow_id": "cfdcacae-a44f-4827-8ad9-db47ad2a8b41",
  "success": true,
  "flow_name": "遍历集合元素测试",
  "message": ""
}
```

2. 从上述响应数据中，记录流运行的running\_id。

**步骤2** 查看流运行记录。

1. 查询流某次运行记录。

- 接口相关信息

URI格式: GET /v2/{project\_id}/flows/{flow\_id}/running-history/{running\_id}

参数说明请参见[查询单个流运行历史记录](#)。

- 请求示例

GET http://{endpoint}/v2/f00ca7a36e\*\*\*\*\*6d6c5f8da8/flows/cfdcacae-a44f-4827-8ad9-db47ad2a8b41/running-history/d96ba061-3caf-49cb-8950-4dac97a8f082

{endpoint}为“mssi.cn-north-4.myhuaweicloud.com”。

- 响应示例

```
{
  "id": "d96ba061-3caf-49cb-8950-4dac97a8f082",
  "flow_id": "cfdcacae-a44f-4827-8ad9-db47ad2a8b41",
  "flow_name": "遍历集合元素测试",
  "run_model": "manual",
  "status": "success",
  "icon": "https://connector-icon.obs.cn-north-7.ulanqab.huawei.com/icon/c9710579264845c6a5339fd738e66fd6/155e6930bdb24d9d95b5d3c636e133eb.png",
  "start_time": 1655881122470,
  "end_time": 1655881126839,
  "exec_time": 4369,
  "message": null,
  "steps": [
    {
      "stepId": "05778538-83a6-467b-bb92-738d9049bb3c",
      "actionName": "手动触发器触发事件",
      "actionType": null,
      "call": {
        "connectorRef": "ManualTrigger",
        "triggerRef": {
          "refName": "手动触发器触发事件",
          "refId": "ManualTrigger",
          "type": "instant",
          "authentication": {
            "connectionId": null,
            "schema": null
          }
        },
        "input": {
          "body": {}
        }
      },
      "config": null,
      "customFunctionMap": null
    },
    {
      "actionRef": null,
      "connectorVersion": "1"
    }
  ],
  "transmit": [
    "bc744b98-007f-496e-bd10-ed460548f7de"
  ],
  "icon": "https://connector-icon.obs.cn-north-7.ulanqab.huawei.com/icon/"
}
```

```
c9710579264845c6a5339fd738e66fd6/155e6930bdb24d9d95b5d3c636e133eb.png",  
  "name": "手动触发流",  
  "lastRunningStatus": "success",  
  "cost": 354  
  }  
]  
}
```

----结束

# 6 权限策略和授权项

## 6.1 权限及授权项说明

使用统一身份认证服务（Identity and Access Management，简称IAM），可以实现对开天集成工作台（MacroVerse SmartStage for Integrators）进行精细的权限管理。如果已经能满足您的要求，不需要创建独立的IAM用户，可以跳过本章节，不影响使用进行精细的权限管理。

默认情况下，新建的IAM用户没有任何权限，需要将其加入用户组，并给用户组授予策略或角色，才能使用户组中的用户获得相应的权限，这一过程称为授权。授权后，用户就可以基于已有权限对云服务进行操作。

权限根据授权的精细程度，分为**角色**和**策略**。角色以服务为粒度，是IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。策略以API接口为粒度进行权限拆分，授权更加精细，可以精确到某个操作、资源和条件，能够满足企业对权限最小化的安全管控要求。

### 📖 说明

如果要允许或是禁止某个接口的操作权限，请使用策略。

账号具备所有接口的调用权限，如果使用账号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来查询流活动历史记录，那么这个IAM用户被授予的策略中必须包含允许“mssi:flow:listRunHistory”的授权项，该接口才能调用成功。

## 支持的授权项

策略包含系统策略和自定义策略，如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝在特定条件下对指定资源进行某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。

- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。
- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。管理员可以在授权项列表中查看授权项是否支持IAM项目或企业项目，“√”表示支持，“×”表示暂不支持。关于IAM项目与企业项目的区别，详情请参见：[IAM与企业管理的区别](#)。

开天集成工作台的支持自定义策略授权项如下所示：

- **流管理**，包括流接口对应的授权项，如提交运行流、启用流程、禁用流程、webhook触发流程调用等接口。

## 6.2 流管理权限

表 6-1 流管理的细化权限说明

权限	对应API接口	授权项	依赖的授权项	IAM项目	企业项目
运行流程	POST /v1/{project_id}/flows/{flow_id}/run	mssi:flow:start	-	√	×
开启流程	POST /v1/{project_id}/flows/{flow_id}/enable	mssi:flow:enable	-	√	×
关闭流程	POST /v1/{project_id}/flows/{flow_id}/disable	mssi:flow:disable	-	√	×
查询流程运行记录	GET /v2/{project_id}/flows/running-history	mssi:flow:listRunHistory	-	√	×
查询单个流程运行记录	GET /v2/{project_id}/flows/{flow_id}/running-history	mssi:flow:listFlowRunHistory	-	√	×
查询流程运行详细记录	GET /v2/{project_id}/flows/{flow_id}/running-history/{running_id}	mssi:flow:getRunHistory	-	√	×



# 7 附录

## 7.1 状态码

状态码如表7-1所示。

表 7-1 状态码

状态码	编码	说明
100	Continue	继续请求。 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。
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。
204	NoContent	请求完全成功，同时HTTP响应不包含响应体。 在响应OPTIONS方法的HTTP请求时返回此状态码。
400	Bad Request	非法请求。 建议直接修改该请求，不要重试该请求。
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。

状态码	编码	说明
403	Forbidden	请求被拒绝访问。 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。 建议直接修改该请求，不要重试该请求。
405	Method Not Allowed	请求中带有该资源不支持的方法。 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。
409	Conflict	服务器在完成请求时发生冲突。 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。
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。
429	Too Many Requests	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	Internal Server Error	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。
503	Service Unavailable	被请求的服务无效。 建议直接修改该请求，不要重试该请求。

## 7.2 错误码

状态码	错误码	错误信息	描述	处理措施
400	MSSI.00010001	connector is not found	连接器不存在	请检查填写的连接器。
400	MSSI.00010002	icon not found	图片资源不存在	请检查填写的图片资源。
400	MSSI.00010003	swagger data not found	swagger资源不存在	请检查填写的swagger资源。
400	MSSI.00010004	connector can not delete	内置连接器无法删除	请正确选择要删除的连接器，内置连接器无法删除。
400	MSSI.00010005	connector was used in flow or template can not delete	该连接器已在流中使用，请删除流后再进行此操作	请先删除使用该连接器的流。
400	MSSI.00010006	connector name can not be null	创建连接器，名称不可为空字符串	请填写连接器名称。
400	MSSI.00011001	flow not found	流不存在	请检查填写的流。
400	MSSI.00011002	flowConnector not found	流对应的连接器找不到	请选择其他连接器。
400	MSSI.00011003	flowRepository not found	流对应的仓库找不到	请选择其他仓库。
400	MSSI.00011004	duplicate flowName	流名称重复	请重命名重复的流名称。
400	MSSI.00012001	template not found	模板不存在	请检查填写的模板。
400	MSSI.00012002	cooperation not found	模板供应商不存在	请选择其他模板供应商。
400	MSSI.00013001	gateway not found	网关不存在	请选择其他网关。
400	MSSI.00014001	share not found	共享数据不存在	请选择其他共享数据。
400	MSSI.00015001	connect not found	连接不存在	请检查填写的连接。

状态码	错误码	错误信息	描述	处理措施
400	MSSI.00015002	connectionName repeat	连接名称重复	请更换重复的名称。
400	MSSI.00016001	function not found	函数找不到	请检查填写的函数。
400	MSSI.00017001	data format is incorrect	数据格式不正确	请检查填写的数据格式。
400	MSSI.00017002	no permission	无权限操作该资源	请添加操作该资源的权限。
400	MSSI.00017003	no certification	无认证方式	请添加认证方式。
400	MSSI.00017004	batch operate apic failed	批量操作APIC失败	请稍后重试，或者联系技术支持人员。
400	MSSI.00017005	operate swagger failed	批量操作SWAGGER失败	请稍后重试，或者联系技术支持人员。
400	MSSI.00017006	to bean failed	JsonUtils	请检查JSON格式的合法性。
400	MSSI.00017007	is not a pic	导入不是图片文件	请导入图片文件。
400	MSSI.00017008	is not a swagger	导入文件不符合Swagger标准	请检查导入的文件是否符合Swagger标准。
400	MSSI.00017010	param is invalid	连接正在使用，无法删除	请先停止正在使用的流程，再删除。
400	MSSI.00017011	flow is enable	流正在使用，无法删除	请先停止正在使用的流，再删除。
400	MSSI.00020001	flow is not exist	流程不存在	请检查填写的流程。
400	MSSI.00020002	connector is not exist	连接器不存在	请检查填写的连接器。
400	MSSI.00020003	connect is not exist	连接不存在（认证连接）	请核对鉴权信息。
400	MSSI.00020005	time_range is invalid, example:2020-11-10T00:00:00.000Zto2022-23-10T00:00:00.000Z	无效的时间范围格式	请更改时间格式为统一UTC时区。

状态码	错误码	错误信息	描述	处理措施
400	MSSI.00020006	decrypt database password error	数据库密码错误	请检查填写的数据库密码。
400	MSSI.00020007	trigger type is invalid.	无效的触发器类型	请检查触发器类型是否符合规范。
400	MSSI.00020008	flow has no trigger step. Please check flow step config.	不存在 FlowStep	请检查FlowStep配置信息。
400	MSSI.00020009	connector is not exist or deploy_url is empty.	连接器不存在或部署地址为空	请检查填写的连接器是否规范，或者是否填写部署地址。
400	MSSI.00020011	param is invalid.	参数无效	请检查填写的参数。
400	MSSI.00020012	running history is not exist.	流运行历史不存在	请检查填写的流。
400	MSSI.00020015	time_range exceeds 30 days.	时间范围超过30天	请检查选择的时间区间
400	MSSI.00020016	connector release version not exists.	连接器发布版本不存在	请检查连接器版本
400	MSSI.00020017	connector invoke record not exists.	连接器调用记录不存在	请检查填写的连接器
400	MSSI.00020018	flow status is disable	流被禁用	请检查流的状态
400	MSSI.00020019	action in not exist	连接器执行动作不存在	请检查执行动作
400	MSSI.00020020	action is not belong to connector	执行动作不属于连接器	请检查执行动作
403	MSSI.00020004	forbidden operation. concat the project owner	操作越权，禁止操作。操作了其他租户的资源	请检查权限信息。

状态码	错误码	错误信息	描述	处理措施
500	MSSI.00029999	unknown error	未知的异常，一般来说是系统内部超出预期的异常	系统错误，请联系技术支持人员。
500	MSSI.00050001	Internal Server Error	未知的异常，一般来说是系统内部超出预期的异常	系统错误，请联系技术支持人员
500	MSSI.00050002	Failed to connection DataSource	连接数据源失败	请检查数据源连接信息
500	MSSI.00050004	Unsupported datasource type	不支持的数据源类型	请检查数据源类型
500	MSSI.00050005	MetaModel adapter is not found	元模型适配器未找到	请检查元模型适配器
500	MSSI.00050003	Task not found	采集任务未找到	请检查采集任务
500	MSSI.00050005	MetaData adapter failed	元数据采集适配器失败	请检查元模型适配器
400	MSSI.00050008	Some params are invalid	参数不合法	请检查参数
400	MSSI.00050009	Resource count over limit	资源超限	请检查资源配额
404	MSSI.00050001	Resource not exist	资源不存在	请检查资源是否存在
403	MSSI.00050002	No permission	没有权限访问	请检查权限
500	MSSI.00050004	Schedule log aspect record schedule log failed	审计日志记录失败	系统错误，请联系技术支持人员
400	MSSI.00050005	Unsupported adapter type	不支持的适配器类型	请检查适配器类型

状态码	错误码	错误信息	描述	处理措施
403	MSSI.00050027	Taskcollector's parameter has modify failed, task maybe in the status of starting, executing and stopfailed.	任务采集失败	请检查采集任务

## 7.3 获取项目 ID

在调用接口的时候，部分请求中需要填入项目ID或项目名称，所以需要获取到项目ID和名称。有如下两种获取方式：

- [从控制台获取项目ID和名称\(推荐使用\)](#)
- [调用API获取项目ID](#)

### 从控制台获取项目 ID 和名称

从控制台获取项目ID（project\_id）和名称（project name）的步骤如下：

- 步骤1** 登录[开天集成工作台管理控制台](#)。
- 步骤2** 鼠标移动至页面右上角账号名上，在下拉列表中单击“我的凭证”。
- 步骤3** 在“API凭证”页面的项目列表中查看项目ID和名称（即“项目”）。

图 7-1 查看项目 ID



----结束

### 调用 API 获取项目 ID

项目ID还可通过调用[查询指定条件下的项目信息](#)API获取。

获取项目ID的接口为GET <https://{iam-endpoint}/v3/projects>，其中{iam-endpoint}为IAM的终端节点，可以从[地区和终端节点](#)处获取。

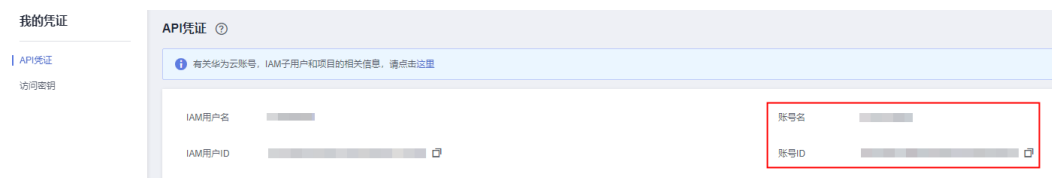
响应体中projects下的“id”即为项目ID。

## 7.4 获取账号 ID

在调用接口的时候，部分请求中需要填入账号名（domain name）和账号ID（domain\_id）。获取步骤如下：

- 步骤1** 注册并登录开天集成工作台控制台。
- 步骤2** 鼠标移动至用户名，在下拉列表中单击“我的凭证”。  
在“API凭证”页面的查看“账号名”和“账号ID”。

图 7-2 获取账号 ID



----结束



# 8 历史 API

## 8.1 查询流历史活动记录

### 说明

当前页面API为历史版本API，未来可能停止维护。请使用[查询流运行历史记录](#)。

### 功能介绍

查询流运行记录。

### URI

GET /v1/{project\_id}/flows/running-history

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，用于资源隔离。获取方式请参考 <a href="#">获取项目ID</a> 。

表 8-2 Query 参数

参数	是否必选	参数类型	描述
time_range	否	String	时间筛选范围，UTC时间。格式：yyyy-MM-dd'T'HH:mm:ss.SSS'Z'toyyyy-MM-dd'T'HH:mm:ss.SSS'Z' 例如： 2020-11-10T00:00:00Zto2020-11-12T00:00:00

参数	是否必选	参数类型	描述
name	否	String	请求名称。
status	否	String	流的状态。 枚举值： <ul style="list-style-type: none"> <li>• <b>success</b></li> <li>• <b>fail</b></li> </ul>
offset	否	Integer	当前页，起始页数应大于等于0。
limit	否	Integer	每页记录数，每页记录数应大于等于1。

## 请求参数

无

## 响应参数

状态码： 200

表 8-3 响应 Body 参数

参数	参数类型	描述
total	Long	历史记录总数。
histories	Array of <b>histories</b> objects	所有历史记录的信息。

表 8-4 histories

参数	参数类型	描述
id	String	流运行记录ID。
flow_id	String	流ID。
flow_name	String	流名称。
flow_version_id	String	流运行版本ID。

参数	参数类型	描述
run_model	String	流运行模式，通常有手动、自动、测试三种。 <ul style="list-style-type: none"> <li>● 手动：需要用户在页面单击运行流。</li> <li>● 自动：会按照触发器的类型自动运行流。</li> <li>● 测试：用户在创建流的时候可以测试流的功能。</li> </ul>
status	String	最后一次运行，流运行状态。 <ul style="list-style-type: none"> <li>● success：流运行成功</li> <li>● fail：流运行失败</li> </ul>
icon	String	流图标。
start_time	String	流运行的开始时间。
end_time	String	流运行的结束时间。
exec_time	Long	执行时间，单位：毫秒。
running_times	Long	运行次数。
last_time	String	最后一次运行时间。
message	String	最后一次运行，输出的错误信息。

**状态码： 403**

**表 8-5 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

**状态码： 500**

表 8-6 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度：8 最大长度：36
error_msg	String	错误描述。 最小长度：2 最大长度：512

## 8.2 查询单个流历史活动记录

### 说明

当前页面API为历史版本API，未来可能停止维护。请使用[查询单个流运行历史记录](#)。

### 功能介绍

根据单个流ID查询运行记录。

### URI

GET /v1/{project\_id}/flows/{flow\_id}/running-history

表 8-7 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，用于资源隔离。获取方式请参考 <a href="#">获取项目ID</a> 。
flow_id	是	String	流程ID。

表 8-8 Query 参数

参数	是否必选	参数类型	描述
time_range	否	String	时间筛选范围，UTC时间。格式：yyyy-MM-dd'T'HH:mm:ss.SSS'Z'toyyyy-MM-dd'T'HH:mm:ss.SSS'Z' 例如： 2020-11-10T00:00:00Zto2020-11-12T00:00:00

参数	是否必选	参数类型	描述
status	否	String	流的状态。 枚举值： <ul style="list-style-type: none"> <li>• <b>success</b></li> <li>• <b>fail</b></li> </ul>
offset	否	Integer	当前页，起始页数应大于等于 0。
limit	否	Integer	每页记录数，每页记录数应大于等于 1。

## 请求参数

无

## 响应参数

状态码： 200

表 8-9 响应 Body 参数

参数	参数类型	描述
total	Long	历史记录总数。
histories	Array of <b>histories</b> objects	所有历史记录的信息。

表 8-10 histories

参数	参数类型	描述
id	String	流运行记录ID。
flow_id	String	流ID。
flow_name	String	流名称。
flow_version_id	String	流运行版本ID。
run_model	String	流运行模式，通常有手动、自动、测试三种。 <ul style="list-style-type: none"> <li>• 手动：需要用户在页面单击运行流。</li> <li>• 自动：会按照触发器的类型自动运行流。</li> <li>• 测试：用户在创建流的时候可以测试流的功能。</li> </ul>

参数	参数类型	描述
status	String	最后一次运行，流程运行状态。 <ul style="list-style-type: none"> <li>• success: 流运行成功</li> <li>• fail: 流运行失败</li> </ul>
icon	String	流图标。
start_time	String	开始时间。
end_time	String	结束时间。
exec_time	Long	执行时间，单位：毫秒。
running_times	Long	运行次数。
last_time	String	最后一次运行时间。
message	String	最后一次运行，输出的错误信息。

**状态码： 403**

**表 8-11 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

**状态码： 500**

**表 8-12 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

## 8.3 获取流运行详细信息

### 📖 说明

当前页面API为历史版本API，未来可能停止维护。请使用[获取流运行详细信息](#)。

### 功能介绍

根据流ID和流运行记录ID查询流运行详细信息，包括每一步的输入、输出、结果、执行时间。

### URI

GET /v1/{project\_id}/flows/{flow\_id}/running-history/{running\_id}

表 8-13 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，用于资源隔离。获取方式请参考 <a href="#">获取项目ID</a> 。
running_id	是	String	流运行记录ID。

### 请求参数

无

### 响应参数

状态码： 200

表 8-14 响应 Body 参数

参数	参数类型	描述
id	String	流运行记录ID。
flow_id	String	流ID。
flow_name	String	流名称。
run_model	String	流运行模式，通常有手动、自动、测试三种。 <ul style="list-style-type: none"><li>● 手动：需要用户在页面单击运行流。</li><li>● 自动：会按照触发器的类型自动运行流。</li><li>● 测试：用户在创建流的时候可以测试流的功能。</li></ul>

参数	参数类型	描述
status	String	最后一次运行，流程运行状态。 <ul style="list-style-type: none"> <li>• success: 流运行成功</li> <li>• fail: 流运行失败</li> </ul>
icon	String	流图标。
start_time	String	开始运行时间。
end_time	String	结束运行时间。
exec_time	Long	执行时间，单位：毫秒。
message	String	最后一次运行，输出的错误信息。
steps	Array of <b>steps</b> objects	一个流通常由多个步骤组成，该数组是流每个步骤的元数据。

表 8-15 steps

参数	参数类型	描述
stepId	String	操作步骤ID。
name	String	连接器名称。
actionName	String	操作步骤名称。
actionType	String	操作步骤类型。
call	<b>call</b> object	连接器操作信息。
icon	String	步骤的图标。
transmit	Array of strings	该步骤执行完之后，下一步操作的ID列表。
mockOutput	String	模拟输出数据。
lastRunningStatus	String	流中该步骤的最后一次运行状态。 <ul style="list-style-type: none"> <li>• success: 流运行成功</li> <li>• fail: 流运行失败</li> </ul>
cost	Long	执行此步骤需要的时间，单位毫秒。

表 8-16 call

参数	参数类型	描述
connectorRef	String	连接器操作ID。



参数	参数类型	描述
triggerRef	<b>triggerRef</b> object	触发器操作定义。
actionRef	<b>actionRef</b> object	普通操作定义。
connectorVersion	String	连接器版本。

表 8-17 triggerRef

参数	参数类型	描述
refName	String	触发器操作名称。
refId	String	触发器操作ID。
type	String	触发器操作类型。
authentication	<b>authentication</b> object	连接器操作认证。
input	Map<String, Object>	连接器执行时的输入数据。
config	Map<String, Object>	连接器配置信息。
customFunctionIds	Array of strings	用户可以在流中添加自定义函数处理数据，该字段表示所有用户自定义的函数ID。
customFunctionMap	Map<String, Object>	自定义函数配置。

表 8-18 authentication

参数	参数类型	描述
connectionId	String	连接器ID。
schema	<b>schema</b> object	连接器认证元数据。

表 8-19 schema

参数	参数类型	描述
id	String	连接ID，流使用连接器时需要配置认证信息，称为连接。

参数	参数类型	描述
authInfo	Map<String,Object>	连接认证信息。
connectorId	String	连接器ID。
type	String	连接器类型。

表 8-20 actionRef

参数	参数类型	描述
refName	String	普通操作名称。
refId	String	普通操作ID。
type	String	普通操作类型。
authentication	<b>authentication</b> object	连接器操作认证。
input	Map<String,Object>	连接器执行时的输入数据。
config	Map<String,Object>	连接器配置信息。
customFunctionIds	Array of strings	用户可以在流中添加自定义函数处理数据，该字段表示所有用户自定义的函数ID。
customFunctionMap	Map<String,Object>	自定义函数配置。

表 8-21 authentication

参数	参数类型	描述
connectionId	String	连接器ID。
schema	<b>schema</b> object	连接器认证元数据。

表 8-22 schema

参数	参数类型	描述
id	String	连接ID。
authInfo	Map<String,Object>	连接认证信息。

参数	参数类型	描述
connectorId	String	连接器ID。
type	String	连接器类型。

状态码： 403

表 8-23 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

状态码： 500

表 8-24 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。 最小长度： 8 最大长度： 36
error_msg	String	错误描述。 最小长度： 2 最大长度： 512

# 9 修订记录

表 9-1 修订记录

发布日期	修订记录
2024-04-24	第六次正式发布。 优化如下章节： <ul style="list-style-type: none"><li>• <a href="#">概述</a></li><li>• <a href="#">终端节点</a></li><li>• <a href="#">API概览</a></li><li>• <a href="#">构造请求</a></li><li>• <a href="#">认证鉴权</a></li><li>• <a href="#">返回结果</a></li><li>• <a href="#">运行流并查看运行记录</a></li><li>• <a href="#">查询流历史活动记录</a></li><li>• <a href="#">查询单个流历史活动记录</a></li><li>• <a href="#">获取流运行详细信息</a></li></ul>
2024-03-11	第五次正式发布。 新增如下章节： <a href="#">应用模型</a> 修改如下章节： <a href="#">API概览</a> ，增加应用模型概览。 <a href="#">错误码</a> ，增加应用模型错误码。
2022-11-28	第四次正式发布。 优化如下章节： <ul style="list-style-type: none"><li>• <a href="#">API概览</a></li><li>• <a href="#">权限及授权项说明</a></li><li>• <a href="#">错误码</a></li></ul>

发布日期	修订记录
2022-11-01	第三次正式发布。 优化如下章节： <ul style="list-style-type: none"><li>● <a href="#">查询流历史活动记录</a></li><li>● <a href="#">查询单个流历史活动记录</a></li><li>● <a href="#">获取流运行详细信息</a></li></ul>
2022-09-29	第二次正式发布。 新增如下章节： <ul style="list-style-type: none"><li>● <a href="#">终端节点</a></li><li>● <a href="#">约束与限制</a></li><li>● <a href="#">基本概念</a></li><li>● <a href="#">应用示例</a></li><li>● <a href="#">历史API</a></li></ul> 修改如下章节： <ul style="list-style-type: none"><li>● 废除<a href="#">webhook触发流程调用（废弃）</a> API。</li><li>● <a href="#">提交运行流</a>，修改请求和响应参数。</li><li>● <a href="#">错误码</a>，修改部分错误码说明。</li></ul>
2022-04-15	第一次正式发布。