

配置审计

# 接口参考

文档版本 01  
发布日期 2023-12-20



版权所有 © 华为技术有限公司 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。

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

# 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 基本概念.....	1
1.5 分页查询.....	2
<b>2 API 概览.....</b>	<b>4</b>
<b>3 如何调用 API.....</b>	<b>5</b>
3.1 构造请求.....	5
3.2 认证鉴权.....	9
3.3 返回结果.....	10
<b>4 API.....</b>	<b>12</b>
4.1 资源清单.....	12
4.1.1 列举指定类型的资源.....	12
4.1.2 列举云服务.....	19
4.1.3 查询单个资源.....	25
4.1.4 列举资源记录器收集的全部资源.....	32
4.1.5 查询资源记录器收集的资源数量.....	38
4.1.6 列举资源记录器收集的资源标签.....	43
4.1.7 列举资源记录器收集的资源概要.....	49
4.1.8 查询资源记录器收集的单个资源.....	54
4.1.9 列举所有资源.....	59
4.1.10 查询账号下的单个资源.....	66
4.1.11 列举资源标签.....	72
4.1.12 查询资源数量.....	78
4.1.13 列举资源概要.....	83
4.2 资源记录器.....	89
4.2.1 查询记录器.....	89
4.2.2 删除记录器.....	95
4.2.3 创建或更新记录器.....	99
4.3 资源关系.....	106
4.3.1 列举资源关系.....	106

4.3.2 列举资源关系详情.....	112
4.4 资源历史.....	118
4.4.1 查询资源历史.....	119
4.5 合规性.....	127
4.5.1 列出内置策略.....	127
4.5.2 查询单个内置策略.....	137
4.5.3 创建合规规则.....	144
4.5.4 列出合规规则.....	152
4.5.5 更新合规规则.....	159
4.5.6 获取单个合规规则.....	168
4.5.7 删除合规规则.....	174
4.5.8 启用合规规则.....	178
4.5.9 停用合规规则.....	183
4.5.10 运行合规评估.....	187
4.5.11 获取规则的评估状态.....	191
4.5.12 获取资源的合规结果.....	196
4.5.13 获取规则的合规结果.....	202
4.5.14 获取用户的合规结果.....	209
4.5.15 更新合规评估结果.....	216
4.5.16 创建组织合规规则.....	223
4.5.17 查询组织合规规则列表.....	231
4.5.18 查询指定组织合规规则.....	238
4.5.19 删除组织合规规则.....	244
4.5.20 更新组织合规规则.....	248
4.5.21 查询组织合规规则部署状态.....	256
4.5.22 查询组织内每个成员账号合规规则部署的详细状态.....	262
4.6 区域管理.....	268
4.6.1 查询用户可见的区域.....	268
4.7 高级查询.....	273
4.7.1 运行高级查询.....	273
4.7.2 创建高级查询.....	281
4.7.3 列举高级查询.....	286
4.7.4 查询高级查询.....	292
4.7.5 更新单个高级查询.....	297
4.7.6 删除高级查询.....	303
4.7.7 列举高级查询 Schema.....	307
4.8 资源聚合器.....	313
4.8.1 创建资源聚合器.....	313
4.8.2 查询资源聚合器列表.....	319
4.8.3 查询指定资源聚合器.....	325
4.8.4 查询指定资源聚合器聚合账号的状态信息.....	330
4.8.5 更新资源聚合器.....	336

4.8.6 删除资源聚合器.....	342
4.8.7 创建资源聚合器授权.....	345
4.8.8 查询资源聚合器授权列表.....	350
4.8.9 删除资源聚合器授权.....	356
4.8.10 查询所有挂起的聚合请求列表.....	359
4.8.11 删除聚合器账号中挂起的授权请求.....	365
4.8.12 查询聚合器中账号资源的计数.....	368
4.8.13 查询聚合器中资源的列表.....	374
4.8.14 查询源账号中资源的详情.....	381
4.8.15 对指定聚合器执行高级查询.....	388
4.8.16 查询聚合器中一个或多个账号的合规概况.....	393
4.8.17 查询聚合合规规则列表.....	399
4.8.18 查询指定聚合合规规则的资源评估结果详情.....	406
4.8.19 查询指定聚合合规规则详情.....	413
4.9 合规规则包.....	420
4.9.1 列举合规规则包.....	420
4.9.2 创建合规规则包.....	426
4.9.3 查看合规规则包.....	433
4.9.4 删除合规规则包.....	439
4.9.5 列举合规规则包的结果概览.....	443
4.9.6 列举合规规则包的评估结果.....	448
4.9.7 列举合规规则包的评估结果详情.....	454
4.9.8 列举合规规则包分数.....	460
4.9.9 列举预定义合规规则包模板.....	466
4.9.10 查看预定义合规规则包模板.....	473
4.9.11 创建组织合规规则包.....	478
4.9.12 列举组织合规规则包.....	484
4.9.13 查看组织合规规则包.....	490
4.9.14 删除组织合规规则包.....	495
4.9.15 查看组织合规规则包部署状态.....	499
4.9.16 查看组织合规规则包部署详细状态.....	505
<b>5 权限和授权项.....</b>	<b>512</b>
5.1 权限策略及授权项说明.....	512
5.2 资源清单.....	513
5.3 资源记录器.....	515
5.4 资源合规.....	515
5.5 高级查询.....	518
5.6 资源聚合器.....	519
5.7 合规规则包.....	521
<b>6 附录.....</b>	<b>525</b>
6.1 错误码.....	525
6.2 支持的服务和资源类型.....	534

---

6.3 获取账号 ID.....	535
6.4 状态码.....	535
6.5 获取项目 ID.....	536
<b>7 修订记录.....</b>	<b>538</b>

# 1 使用前必读

## 1.1 概述

配置审计（Config）服务提供全局资源配置的检索，配置历史追溯，以及基于资源配置的持续的审计评估能力，确保云上资源配置变更符合客户预期。

本文档提供了配置审计服务API的描述、参数说明以及示例等内容。您可以使用本文档提供的API对Config进行相关操作。支持的全部操作请参阅[API概览](#)。

在调用Config的API前，请确保已经充分了解Config的相关概念，详细信息请参阅[产品介绍](#)。

## 1.2 调用说明

Config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](#)。

## 1.3 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，您可以从[地区和终端节点](#)中查询配置审计服务的终端节点。

## 1.4 基本概念

- 组织

组织云服务为企业用户提供多账号关系的管理能力。组织支持用户将多个华为云账号整合到创建的组织中，并可以集中管理组织下的所有账号。用户可以在组织中设置访问策略，帮助用户更好地满足业务的安全性和合规性需求。

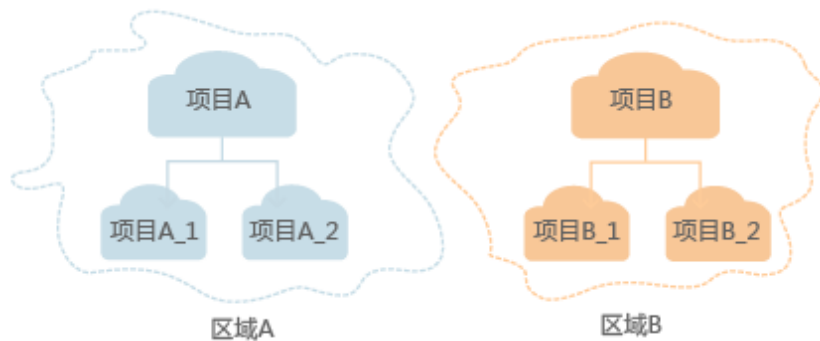
- 账号

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他们进行日常管理工作。



- 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[我的凭证](#)下，您可以查看账号ID和IAM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。
- 区域（Region）  
从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。  
详情请参见[区域和可用区](#)。
- 可用区（AZ，Availability Zone）  
一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。
- 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的资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



同样在[我的凭证](#)下，您可以查看项目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的资源进行分组和管理，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《[企业管理用户指南](#)》。

## 1.5 分页查询

配置审计服务的部分API提供分页查询功能，通过在请求的url中添加limit和marker参数实现分页返回列表信息，marker的值必须是上次分页查询返回的值。

表 1-1 配置审计服务的分页查询参数列表

参数名称	类型	必选	说明
limit	Integer	否	每次分页查询返回的最大数量，当limit参数非法时，response将响应错误码400。
marker	String	否	上一次分页查询返回的marker值，当marker值非法时，response将响应错误码400。

# 2 API 概览

表 2-1 配置审计服务接口列表

类型	说明
资源查询	包括查询资源、查询资源标签、查询资源数量和概要、查询资源记录器收集的 资源、列举云服务接口。
资源记录器	包括资源记录器的增、删、改、查接口。
资源关系	查询资源关系和详情接口。
资源历史	查询资源历史接口。
合规性	包括列举和查询合规策略，合规规则的增、删、改、查，运行合规评估，获取和更新合规结果，组织合规规则的增、删、改、查等接口。
区域管理	查询用户可见的区域接口。
高级查询	包括运行高级查询，高级查询的增、删、改、查等接口。
资源聚合器	包括资源聚合器的增、删、改、查，资源聚合器授权的增、删、改、查，查询资源聚合器中的资源计数，以及对资源聚合器运行高级查询等接口。
合规规则包	包括合规规则包和组织合规规则包的增、删、改、查接口，以及合规性数据的查询接口。

# 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](#)来说明如何调用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。

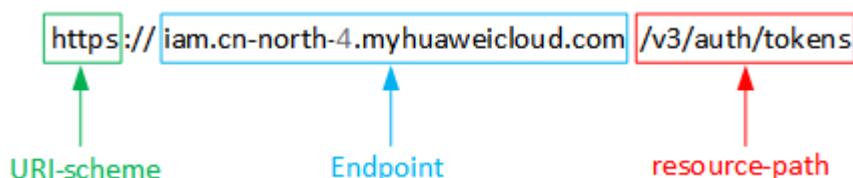
表 3-1 URI 中的参数说明

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，您可以从 <a href="#">地区和终端节点</a> 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，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-北京四”区域的Token，则需使用“华北-北京四”区域的Endpoint（iam.cn-north-4.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-4.myhuaweicloud.com/v3/auth/tokens
```

图 3-1 URI 示意图



### 说明

为方便查看，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。

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
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认证时该字段必选。	code.test.com or code.test.com:443
Content-Type	消息体的类型（格式）。推荐用户使用默认值application/json，有其他取值时会在具体接口中专门说明。	是	application/json
Content-Length	请求body长度，单位为Byte。	否	3495
X-Project-Id	project id，项目编号。请参考 <a href="#">获取项目ID</a> 章节获取项目编号。	否 如果是专属云场景采用AK/SK认证方式的接口请求，或者多project场景采用AK/SK认证的接口请求，则该字段必选。	e9993fc787d94b6c886cb aa340f9c0f4
X-Auth-Token	用户Token。 用户Token也就是调用 <a href="#">获取用户Token</a> 接口的响应值，该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（Headers）中包含的“X-Subject-Token”的值即为Token值。	否 使用Token认证时该字段必选。	注：以下仅为Token示例片段。 MIIPAgYJKoZlhvcNAQcCo ...ggg1BBIIlNPXsidG9rZ

## 📖 说明

API同时支持使用AK/SK认证，AK/SK认证使用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[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://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

## 请求消息体（可选）

该部分可选。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[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，***\$ADMIN\_PASS***表示用户登录密码，***xxxxxxxxxxxxxxxxxxxx***为project的名称，如“cn-north-1”，您可以从[地区和终端节点](#)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的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": "$ADMIN_PASS", //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
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

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的

“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，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证调用请求。
- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。推荐使用AK/SK认证，其安全性比Token认证要高。

### 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。

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加到请求消息头中，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。

云服务存在两种部署方式：项目级服务和全局级服务。

- 项目级服务需要获取项目级别的Token，此时请求body中**auth.scope**的取值为**project**。
- 全局级服务需要获取全局级别的Token，此时请求body中**auth.scope**的取值为**domain**。

调用配置审计服务API需要全局级别的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**domain**，如下所示。

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username", //IAM用户名
          "password": "*****", //IAM用户密码
          "domain": {
            "name": "domainname" //IAM用户所属账号名
          }
        }
      }
    },
    "scope": {
      "domain": {
        "name": "xxxxxxx" //租户名称
      }
    }
  }
}
```

获取Token后，再调用其他接口时，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。

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/projects
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在12MB以内，12MB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 )：私有访问密钥。与访问密钥ID结合使用，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，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](#)。

### 说明

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。

## 3.3 返回结果

### 状态码

请求发送以后，您会收到响应，其中包含状态码、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[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。

### 说明

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。

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 → [REDACTED]
x-xss-protection → 1; mode=block;
```

## 响应消息体（可选）

该部分可选。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[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": "az-01",
            .....

```

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 request message format is invalid.",
  "error_code": "IMG.0001"
}
```

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

# 4 API

## 4.1 资源清单

### 4.1.1 列举指定类型的资源

#### 功能介绍

返回当前用户下特定资源类型的资源，需要当前用户有rms:resources:list权限。比如查询云服务器，对应的资源类型是ecs.cloudservers，其中provider为ecs，type为cloudservers。支持的云服务（provider）和资源类型（type）请参见“附录-支持的服务和资源类型”章节。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/provider/{provider}/type/{type}/resources

表 4-1 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
provider	是	String	云服务名称。 最大长度：20
type	是	String	资源类型名称。 最大长度：32

表 4-2 Query 参数

参数	是否必选	参数类型	描述
region_id	否	String	区域ID。 最大长度：36
ep_id	否	String	企业项目ID。 最大长度：36
tag	否	Map<String,String>	标签。
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-3 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-4 响应 Body 参数

参数	参数类型	描述
resources	Array of <a href="#">ResourceEntity</a> objects	资源列表。

参数	参数类型	描述
page_info	<b>PageInfo</b> object	分页对象。

表 4-5 ResourceEntity

参数	参数类型	描述
id	String	资源ID。
name	String	资源名称。
provider	String	云服务名称。
type	String	资源类型。
region_id	String	区域ID。
project_id	String	Openstack中的项目ID。
project_name	String	Openstack中的项目名称。
ep_id	String	企业项目ID。
ep_name	String	企业项目名称。
checksum	String	资源详情校验码。
created	String	资源创建时间。
updated	String	资源更新时间。
provisioning_state	String	资源操作状态。枚举值：Succeeded（资源操作成功）、Failed（资源操作异常）、Canceled（资源操作取消）、Processing（资源操作正在进行中）。
tags	Map<String,String>	资源标签。
properties	Map<String,Object>	资源详细属性。

表 4-6 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>

参数	参数类型	描述
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-7 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-8 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-9 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询当前用户下所有虚拟机。

GET https://{endpoint}/v1/resource-manager/domains/{domain\_id}/provider/ecs/type/cloudServers/resources

## 响应示例

状态码：200

操作成功。

```
{
  "page_info": {
    "current_count": 2,
    "next_marker": null
  },
  "resources": [ {
    "checksum": "89ca775e88e04b2c200ccbf9e219ad0d7da42e3f446e5c953d443288134eec41",
    "created": "2020-02-21T08:41:05Z",
    "ep_id": "0",
    "ep_name": "default",
    "id": "7ffd8564-d88a-4bc9-ab51-d8b79a57d0e6",
    "name": "ecs-test-1",
    "project_id": "059b5e0a2500d5552fa1c00adada8c06",
    "project_name": "project_name",
    "properties": {
      "status": "ACTIVE"
    },
    "provider": "ecs",
    "provisioning_state": "Succeeded",
    "region_id": "regionid1",
    "tags": {
      "use": "test"
    },
    "type": "cloudServers",
    "updated": "2020-02-21T08:41:05Z"
  }, {
    "checksum": "db2aad42804951c03a724b7501da9b6b4c14d319dd319d76bb7c658f256a37b0",
    "created": "2020-02-24T08:43:05Z",
    "ep_id": "0",
    "ep_name": "default",
    "id": "b63b33b7-f48c-4048-995b-0445d124a445",
    "name": "ecs-test-2",
    "project_id": "059b5e0a2500d5552fa1c00adada8c06",
    "project_name": "project_name_1",
    "properties": {
      "status": "ACTIVE"
    },
    "provider": "ecs",
    "provisioning_state": "Succeeded",
    "region_id": "regionid1",
    "tags": {
      "use": "test1"
    },
    "type": "cloudServers",
    "updated": "2020-08-11T11:55:08Z"
  } ]
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListResourcesSolution {
```

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

    ICredential auth = new GlobalCredentials()
        .withAk(ak)
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    ListResourcesRequest request = new ListResourcesRequest();
    request.withRegionId("<region_id>");
    request.withEpId("<ep_id>");
    request.withTag("<tag>");
    request.withLimit(<limit>);
    request.withMarker("<marker>");
    try {
        ListResourcesResponse response = client.listResources(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListResourcesRequest()
        request.region_id = "<region_id>"
        request.ep_id = "<ep_id>"
```



```
request.tag = "<tag>"
request.limit = <limit>
request.marker = "<marker>"
response = client.list_resources(request)
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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListResourcesRequest{}
    regionIdRequest := "<region_id>"
    request.RegionId = &regionIdRequest
    epIdRequest := "<ep_id>"
    request.Epld = &epIdRequest
    tagRequest := "<tag>"
    request.Tag = &tagRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListResources(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.1.2 列举云服务

### 功能介绍

查询支持的云服务、资源和区域列表。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/providers

表 4-10 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-11 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页偏移。 最小值：1 最大值：1000

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
track	否	String	资源是否默认收集，"tracked"表示默认收集，"untracked"表示默认不收集。

## 请求参数

表 4-12 请求 Header 参数

参数	是否必选	参数类型	描述
X-Language	否	String	选择接口返回的信息的语言，默认为"zh-cn"中文，可选值：zh-cn和en-us。 缺省值：zh-cn
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-13 响应 Body 参数

参数	参数类型	描述
resource_providers	Array of <a href="#">ResourceProviderResponse</a> objects	云服务详情列表。
total_count	Integer	当前支持的云服务总数。

表 4-14 ResourceProviderResponse

参数	参数类型	描述
provider	String	云服务名称。
display_name	String	云服务显示名称，可以通过请求Header中的'X-Language'设置语言。
category_display_name	String	云服务类别显示名称，可以通过请求Header中的'X-Language'设置语言。
resource_types	Array of ResourceTypesResponse objects	资源类型列表。

表 4-15 ResourceTypeResponse

参数	参数类型	描述
name	String	资源类型名称。
display_name	String	资源类型显示名称，可以通过请求中'X-Language'设置语言。
global	Boolean	是否是全局类型的资源。
regions	Array of strings	支持的region列表。
console_endpoint_id	String	console终端ID。
console_list_url	String	console列表页地址。
console_detail_url	String	console详情页地址。
track	String	资源是否默认搜集，"tracked"表示默认收集，"untracked"表示默认不收集。

状态码： 500

表 4-16 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "total_count": 2,
  "resource_providers": [ {
    "provider": "ecs",
    "display_name": "弹性云服务器 ECS",
    "category_display_name": "计算",
    "resource_types": [ {
      "name": "cloudservers",
      "display_name": "云服务器",
      "global": false,
      "regions": [ "regionid1", "regionid2", "regionid3", "regionid4", "regionid5", "regionid6" ],
      "console_endpoint_id": "ecm",
      "console_list_url": "#/ecs/manager/vmList",
      "console_detail_url": "#/ecs/manager/ecsDetail?instanceId={id}",
      "track": "tracked"
    } ]
  } ], {
    "provider": "vpc",
    "display_name": "虚拟私有云 VPC",
    "category_display_name": "网络",
    "resource_types": [ {
      "name": "vpcs",
      "display_name": "虚拟私有云",
      "global": false,
      "regions": [ "regionid1", "regionid2", "regionid3", "regionid4", "regionid5", "regionid6" ],
      "console_endpoint_id": "vpc",
      "console_list_url": "#/vpcs",
      "console_detail_url": "#/vpc/vpcmanager/vpcDetail/subnets?vpcId={id}",
      "track": "tracked"
    } ], {
      "name": "bandwidths",
      "display_name": "共享带宽",
      "global": false,
      "regions": [ "regionid1", "regionid2", "regionid3", "regionid4", "regionid5", "regionid6" ],
      "console_endpoint_id": "vpc",
      "console_list_url": "#/vpc/vpcmanager/shareBandwidth",
      "console_detail_url": "#/vpc/vpcmanager/shareBandwidth?bandwidthId={id}",
      "track": "tracked"
    } ], {
      "name": "securityGroups",
      "display_name": "安全组",
      "global": false,
      "regions": [ "regionid1", "regionid2", "regionid5", "regionid6" ],
      "console_endpoint_id": "vpc",
      "console_list_url": "#/secGroups",
      "console_detail_url": "#/vpc/vpcmanager/sgDetail/sgRules?instanceId={id}",
      "track": "tracked"
    } ], {
      "name": "publicips",
      "display_name": "弹性公网IP",
      "global": false,
      "regions": [ "regionid1", "regionid2", "regionid3", "regionid4", "regionid6" ],
      "console_endpoint_id": "vpc",
      "console_list_url": "#/vpc/vpcmanager/eips",
      "console_detail_url": "#/vpc/vpcmanager/eipDetailNew?eipId={id}",
      "track": "tracked"
    } ]
  } ]
}
```

```
    }  
  }  
}
```

## 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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class ListProvidersSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListProvidersRequest request = new ListProvidersRequest();  
        request.withTrack(ListProvidersRequest.TrackEnum.fromValue("<track>"));  
        request.withOffset("<offset>");  
        request.withLimit("<limit>");  
        try {  
            ListProvidersResponse response = client.listProviders(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } 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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
```

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

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListProvidersRequest()
        request.track = "<track>"
        request.offset = <offset>
        request.limit = <limit>
        response = client.list_providers(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListProvidersRequest{}
    trackRequest:= model.GetListProvidersRequestTrackEnum().<TRACK>
    request.Track = &trackRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
```

```
limitRequest:= int32(<limit>)  
request.Limit = &limitRequest  
response, err := client.ListProviders(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.1.3 查询单个资源

### 功能介绍

指定资源ID，返回该资源的详细信息，需要当前用户有rms:resources:get权限。比如查询云服务器，对应的资源类型是ecs.cloudservers，其中provider为ecs，type为cloudservers。支持的云服务（provider）和资源类型（type）请参见“附录-支持的服务和资源类型”章节。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/provider/{provider}/type/{type}/resources/{resource\_id}

表 4-17 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36



参数	是否必选	参数类型	描述
provider	是	String	云服务名称。 最大长度：20
type	是	String	资源类型名称。 最大长度：32
resource_id	是	String	资源ID。 最大长度：512

## 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	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-19 响应 Body 参数

参数	参数类型	描述
id	String	资源ID。
name	String	资源名称。
provider	String	云服务名称。
type	String	资源类型。
region_id	String	区域ID。
project_id	String	Openstack中的项目ID。
project_name	String	Openstack中的项目名称。
ep_id	String	企业项目ID。
ep_name	String	企业项目名称。
checksum	String	资源详情校验码。

参数	参数类型	描述
created	String	资源创建时间。
updated	String	资源更新时间。
provisioning_state	String	资源操作状态。枚举值：Succeeded（资源操作成功）、Failed（资源操作异常）、Canceled（资源操作取消）、Processing（资源操作正在进行中）。
tags	Map<String,String>	资源标签。
properties	Map<String,Object>	资源详细属性。

状态码： 400

表 4-20 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-21 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-22 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-23 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

GET https://{endpoint}/v1/resource-manager/domains/059b5c937100d3e40ff0c00a7675a0a0/provider/ecs/type/cloudservers/resources/00337e93-82d1-40ca-911f-07cff94587cc

## 响应示例

状态码： 200

操作成功。

```
{
  "id": "00337e93-82d1-40ca-911f-07cff94587cc",
  "name": "dev_machine",
  "provider": "ecs",
  "type": "cloudservers",
  "region_id": "regionid4",
  "project_id": "39c2af998c334ed6bcbb75b27318f7cc",
  "project_name": "project_name",
  "ep_id": "0",
  "ep_name": "default",
  "checksum": "3a0075409edb156a74e041b7479f0d5993be1d62b4ccd2af3a1dd01ec80c8b39",
  "created": "2019-11-20T06:24:43Z",
  "updated": "2020-07-17T08:30:52Z",
  "provisioning_state": "Succeeded",
  "tags": {
    "usage": "展示"
  },
  "properties": {
    "accessIpv4": "",
    "hostName": "dev-machine",
    "addresses": [ {
      "OsExtIpsType": "fixed",
      "OsExtIpsPortId": "f2fa750a-e2ab-434f-b14a-bfe7c8cea0cc",
      "addr": "192.168.1.212",
      "version": 4,
      "OsExtIpsMacAddr": "fa:16:3e:6e:cf:33"
    }, {
      "OsExtIpsType": "floating",
      "OsExtIpsPortId": "f2fa750a-e2ab-434f-b14a-bfe7c8cea0cc",
      "addr": "100.85.225.33",
      "version": 4,
      "OsExtIpsMacAddr": "fa:16:3e:6e:cf:33"
    } ],
    "accessIpv6": "",
    "metadata": {
      "chargingMode": "0",
      "meteringImageType": "private",
      "imageName": "resource-manager-devmachine-template",
      "meteringImageId": "9bcaace4-b8da-4008-a352-3f72e1f25333",
      "meteringResourcesPerCode": "si2.large.2.linux",
      "vpclId": "cf403ef5-90df-4e7e-829d-5d21b1cb7d1e",
      "osBit": "64"
    },
    "OsExtStsVmState": "active",
    "configDrive": "",
    "OsExtStsPowerState": 1,
  }
}
```

```
"hostId" : "3c381dcfc3e628c1a504ad94ba8c4e89081306455273701333f32921",
"securityGroup" : [ {
  "name" : "default",
  "id" : "5d55b397-ad9c-462d-af72-6599cb941c49"
} ],
"ExtVolumesAttached" : [ {
  "bootIndex" : "0",
  "id" : "010d940e-a73e-417b-85ae-51b76c0d2ba0",
  "device" : "/dev/vda"
} ],
"userId" : "e311190745e94cc09d62d5779e55912d",
"flavor" : {
  "disk" : "0",
  "name" : "Si2.large.2",
  "id" : "Si2.large.2",
  "vcpus" : "2",
  "ram" : "4096"
},
"OsDcfDiskConfig" : "MANUAL",
"hostStatus" : "UP",
"OsSrvUsgLaunchedAt" : "2019-11-20T06:24:56.000000",
"OsExtAz" : "regionid4a",
"progress" : 0,
"locked" : false,
"OS-EXT-SRV-ATTR" : {
  "hostName" : "dev-machine",
  "kernelId" : "",
  "ramdiskId" : "",
  "reservationId" : "r-hhux9o7m",
  "instanceName" : "instance-0009cb50",
  "host" : "regionid4a-pod01.regionid4",
  "rootDeviceName" : "/dev/vda",
  "hypervisorHostName" : "nova001@2",
  "launchIndex" : 0
},
"status" : "ACTIVE"
}
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowResourceByIdSolution {

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

```
ICredential auth = new GlobalCredentials()
    .withAk(ak)
    .withSk(sk);

ConfigClient client = ConfigClient.newBuilder()
    .withCredential(auth)
    .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
    .build();
ShowResourceByIdRequest request = new ShowResourceByIdRequest();
try {
    ShowResourceByIdResponse response = client.showResourceById(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowResourceByIdRequest()
        response = client.show_resource_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
)
```

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.1.4 列举资源记录器收集的全部资源

### 功能介绍

查询当前用户资源记录器收集的全部资源，需要当前用户有rms:resources:list权限。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/tracked-resources

表 4-24 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-25 Query 参数

参数	是否必选	参数类型	描述
region_id	否	String	区域ID。 最大长度：36
ep_id	否	String	企业项目ID。 最大长度：36
type	否	String	资源类型列表，资源类型的格式为provider.type。 最大长度：40
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：100
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
id	否	String	资源ID。 最大长度：512

参数	是否必选	参数类型	描述
name	否	String	资源名称。 最大长度：256
tags	否	Array	标签列表, 标签值的格式为key或key=value。 数组长度：1 - 5

## 请求参数

无

## 响应参数

状态码：200

表 4-26 响应 Body 参数

参数	参数类型	描述
resources	Array of ResourceEntity objects	资源列表。
page_info	PageInfo object	分页对象。

表 4-27 ResourceEntity

参数	参数类型	描述
id	String	资源ID。
name	String	资源名称。
provider	String	云服务名称。
type	String	资源类型。
region_id	String	区域ID。
project_id	String	Openstack中的项目ID。
project_name	String	Openstack中的项目名称。
ep_id	String	企业项目ID。
ep_name	String	企业项目名称。
checksum	String	资源详情校验码。



参数	参数类型	描述
created	String	资源创建时间。
updated	String	资源更新时间。
provisioning_state	String	资源操作状态。枚举值：Succeeded（资源操作成功）、Failed（资源操作异常）、Canceled（资源操作取消）、Processing（资源操作正在进行中）。
tags	Map<String,String>	资源标签。
properties	Map<String,Object>	资源详细属性。

表 4-28 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码：400

表 4-29 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-30 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-31 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

- 查询当前用户下所有资源。

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/tracked-resources
```

- 查询当前用户default企业项目下的资源，返回前100条。

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/tracked-resources?  
limit=100&ep_id=0
```

## 响应示例

状态码： 200

操作成功

```
{  
  "resources": [ {  
    "id": "3ccd9191-6a5e-4939-a971-4652db18b370",  
    "name": "elb-265a",  
    "provider": "elb",  
    "type": "loadbalancers",  
    "region_id": "regionid1",  
    "project_id": "05498e12458025102ff5c0061a584a9f",  
    "project_name": "regionid1_region_service",  
    "ep_id": "0",  
    "ep_name": "default",  
    "checksum": "6e0271b107b764b19acb235f45c0d852f72104fe1d4b32970686e7eae8e87bf4",  
    "created": "2020-02-29T09:39:19Z",  
    "updated": "2020-02-29T09:39:19Z",  
    "provisioning_state": "Succeeded",  
    "tags": { },  
    "properties": {  
      "tenant_id": "05498e12458025102ff5c0061a584a9f",  
      "listeners": [ {  
        "id": "37de3be0-1803-43e2-9bb5-243b4b30b771"  
      } ],  
      "provisioning_status": "ACTIVE",  
      "description": "",  
      "pools": [ {  
        "id": "a7ae4e7f-3e97-48ea-bdbc-334684aafac4"  
      } ],  
      "created_at": "2020-02-29T09:39:19"  
    }  
  } ],  
  "page_info": {  
    "current_count": 1,  
    "next_marker": "CAESJhIkNGEzNWYxMmYtZjM0NC00YTnkLWE1NDEtZWVjMDBiNDIzMjg1GgTGWJss"  
  }  
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

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

public class ListTrackedResourcesSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

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

        ListTrackedResourcesRequest request = new ListTrackedResourcesRequest();
        request.withRegionId("<region_id>");
        request.withEpld("<ep_id>");
        request.withType("<type>");
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        request.withId("<id>");
        request.withName("<name>");
        request.withTags();
        try {
            ListTrackedResourcesResponse response = client.listTrackedResources(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListTrackedResourcesRequest()
        request.region_id = "<region_id>"
        request.ep_id = "<ep_id>"
        request.type = "<type>"
        request.limit = <limit>
        request.marker = "<marker>"
        request.id = "<id>"
        request.name = "<name>"
        request.tags =
        response = client.list_tracked_resources(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.ListTrackedResourcesRequest{  
    regionIdRequest:= "<region_id>"  
    request.RegionId = &regionIdRequest  
    epIdRequest:= "<ep_id>"  
    request.EpId = &epIdRequest  
    typeRequest:= "<type>"  
    request.Type = &typeRequest  
    limitRequest:= int32(<limit>)  
    request.Limit = &limitRequest  
    markerRequest:= "<marker>"  
    request.Marker = &markerRequest  
    idRequest:= "<id>"  
    request.Id = &idRequest  
    nameRequest:= "<name>"  
    request.Name = &nameRequest  
    response, err := client.ListTrackedResources(request)  
    if err == nil {  
        fmt.Printf("%v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功
400	输入参数不合法
403	用户身份认证失败
500	服务器内部错误

## 错误码

请参见[错误码](#)。

## 4.1.5 查询资源记录器收集的资源数量

### 功能介绍

查询当前用户资源记录器收集的资源数量。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/tracked-resources/count

表 4-32 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-33 Query 参数

参数	是否必选	参数类型	描述
id	否	String	资源ID。 最大长度：512
name	否	String	资源名称。 最大长度：256
type	否	Array	资源类型列表，资源类型的格式为provider.type。 数组长度：1 - 100
region_id	否	Array	区域ID列表。 数组长度：1 - 10
ep_id	否	Array	企业项目ID列表。 数组长度：1 - 10
project_id	否	Array	项目ID。 数组长度：1 - 10
tags	否	Array	标签列表，标签值的格式为key或key=value。 数组长度：1 - 5

## 请求参数

无

## 响应参数

状态码：200

表 4-34 响应 Body 参数

参数	参数类型	描述
total_count	Integer	资源总数。

状态码： 400

表 4-35 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-36 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-37 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询当前用户资源记录器收集的资源数量。

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/tracked-resources/count
```

## 响应示例

状态码： 200

操作成功

```
{  
  "total_count" : 345  
}
```

## 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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CountTrackedResourcesSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CountTrackedResourcesRequest request = new CountTrackedResourcesRequest();  
        request.withId("<id>");  
        request.withName("<name>");  
        request.withType();  
        request.withRegionId();  
        request.withEpld();  
        request.withProjectId();  
        request.withTags();  
        try {  
            CountTrackedResourcesResponse response = client.countTrackedResources(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CountTrackedResourcesRequest()
        request.id = "<id>"
        request.name = "<name>"
        request.type =
        request.region_id =
        request.ep_id =
        request.project_id =
        request.tags =
        response = client.count_tracked_resources(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
```

```
config.ConfigClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build())

request := &model.CountTrackedResourcesRequest{}
idRequest:= "<id>"
request.Id = &idRequest
nameRequest:= "<name>"
request.Name = &nameRequest
response, err := client.CountTrackedResources(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功
400	输入参数不合法
403	用户认证失败
500	服务器内部错误

## 错误码

请参见[错误码](#)。

### 4.1.6 列举资源记录器收集的资源标签

#### 功能介绍

查询当前用户资源记录器收集的资源的标签。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/tracked-resources/tags

表 4-38 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-39 Query 参数

参数	是否必选	参数类型	描述
key	否	String	标签键名。 最大长度：128
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：100

## 请求参数

无

## 响应参数

状态码：200

表 4-40 响应 Body 参数

参数	参数类型	描述
tags	Array of <b>TagEntity</b> objects	标签列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-41 TagEntity

参数	参数类型	描述
key	String	标签键。
value	Array of strings	标签键对应的标签值列表。

表 4-42 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-43 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-44 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-45 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询当前用户资源记录器收集的资源的标签。

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/tracked-resources/tags
```

## 响应示例

**状态码： 200**

操作成功

```
{
  "tags": [ {
    "key": "chloe",
    "value": [ "a", "b" ]
  } ]
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListTrackedResourceTagsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
ListTrackedResourceTagsRequest request = new ListTrackedResourceTagsRequest();
request.withKey("<key>");
request.withMarker("<marker>");
request.withLimit(<limit>);
try {
    ListTrackedResourceTagsResponse response = client.listTrackedResourceTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

    client = ConfigClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ConfigRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = ListTrackedResourceTagsRequest()
        request.key = "<key>"
        request.marker = "<marker>"
        request.limit = <limit>
        response = client.list_tracked_resource_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListTrackedResourceTagsRequest{}  
    keyRequest := "<key>"  
    request.Key = &keyRequest  
    markerRequest := "<marker>"  
    request.Marker = &markerRequest  
    limitRequest := int32(<limit>)  
    request.Limit = &limitRequest  
    response, err := client.ListTrackedResourceTags(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功
400	输入参数不合法
403	用户认证失败
500	服务器内部错误

## 错误码

请参见[错误码](#)。

## 4.1.7 列举资源记录器收集的资源概要

### 功能介绍

查询当前用户资源记录器收集的资源概览。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/tracked-resources/summary

表 4-46 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-47 Query 参数

参数	是否必选	参数类型	描述
name	否	String	资源名称。 最大长度：256
type	否	Array	资源类型列表，资源类型的格式为provider.type。 数组长度：1 - 100
region_id	否	Array	区域ID列表。 数组长度：1 - 10
ep_id	否	Array	企业项目ID列表。 数组长度：1 - 10
project_id	否	Array	项目ID。 数组长度：1 - 10
tags	否	Array	标签列表，标签值的格式为key或key=value。 数组长度：1 - 5

### 请求参数

无



## 响应参数

状态码： 200

表 4-48 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of <b>ResourceSummaryResponseItem</b> objects	资源概要信息列表。

表 4-49 ResourceSummaryResponseItem

参数	参数类型	描述
provider	String	云服务名称。
types	Array of <b>types</b> objects	资源类型列表。

表 4-50 types

参数	参数类型	描述
type	String	资源类型名称。
regions	Array of <b>regions</b> objects	区域列表。

表 4-51 regions

参数	参数类型	描述
region_id	String	区域ID。
count	Long	该资源类型在当前区域的数量。

状态码： 400

表 4-52 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	错误消息内容。

状态码： 403

表 4-53 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-54 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询当前用户资源记录器收集的资源概览。

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/tracked-resources/summary
```

## 响应示例

状态码： 200

Successful Operation

```
[{  
  "provider": "ecs",  
  "types": [{  
    "type": "buckets",  
    "regions": [{  
      "region_id": "regionid1",  
      "count": 5  
    }]  
  }]  
}]
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

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

public class CollectTrackedResourcesSummarySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        CollectTrackedResourcesSummaryRequest request = new CollectTrackedResourcesSummaryRequest();
        request.withName("<name>");
        request.withType();
        request.withRegionId();
        request.withEpld();
        request.withProjectId();
        request.withTags();
        try {
            CollectTrackedResourcesSummaryResponse response =
client.collectTrackedResourcesSummary(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *
```

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CollectTrackedResourcesSummaryRequest()
        request.name = "<name>"
        request.type =
        request.region_id =
        request.ep_id =
        request.project_id =
        request.tags =
        response = client.collect_tracked_resources_summary(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CollectTrackedResourcesSummaryRequest{}
    nameRequest := "<name>"
    request.Name = &nameRequest
    response, err := client.CollectTrackedResourcesSummary(request)
    if err == nil {
```

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

## 更多

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

## 状态码

状态码	描述
200	Successful Operation
400	Invalid Param Supplied
403	Authentication Failed
500	Internal Error

## 错误码

请参见[错误码](#)。

## 4.1.8 查询资源记录器收集的单个资源

### 功能介绍

查询当前用户资源记录器收集的单个资源。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/tracked-resources/{resource\_id}

表 4-55 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
resource_id	是	String	资源ID。 最大长度：512

## 请求参数

无

## 响应参数

状态码： 200

表 4-56 响应 Body 参数

参数	参数类型	描述
id	String	资源ID。
name	String	资源名称。
provider	String	云服务名称。
type	String	资源类型。
region_id	String	区域ID。
project_id	String	Openstack中的项目ID。
project_name	String	Openstack中的项目名称。
ep_id	String	企业项目ID。
ep_name	String	企业项目名称。
checksum	String	资源详情校验码。
created	String	资源创建时间。
updated	String	资源更新时间。
provisioning_state	String	资源操作状态。枚举值：Succeeded（资源操作成功）、Failed（资源操作异常）、Canceled（资源操作取消）、Processing（资源操作正在进行中）。
tags	Map<String,String>	资源标签。
properties	Map<String,Object>	资源详细属性。

状态码： 400

表 4-57 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

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	错误消息内容。

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	错误消息内容。

## 请求示例

查询当前用户资源记录器收集的单个资源。

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/tracked-resources/{resource_id}
```

## 响应示例

状态码： 200

操作成功

```
{
  "id": "00337e93-82d1-40ca-911f-07cff94587cc",
  "name": "dev_machine",
  "provider": "vpc",
  "type": "vpcs",
  "region_id": "regionid1",
  "project_id": "39c2af998c334ed6bcbb75b27318f7cc",
  "project_name": "regionid1",
  "ep_id": "0",
  "ep_name": "default",
  "checksum": "3a0075409edb156a74e041b7479f0d5993be1d62b4ccd2af3a1dd01ec80c8b39",
  "created": "2019-11-20T06:24:43Z",
  "updated": "2020-07-17T08:30:52Z",
  "provisioning_state": "Succeeded",
  "tags": {
    "usage": "mm1"
  },
  "properties": {
    "cidr": "192.168.0.0/16",
    "status": "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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowTrackedResourceDetailSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTrackedResourceDetailRequest request = new ShowTrackedResourceDetailRequest();
        try {
            ShowTrackedResourceDetailResponse response = client.showTrackedResourceDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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



```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = ShowTrackedResourceDetailRequest()
    response = client.show_tracked_resource_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	操作成功
400	输入参数不合法
403	用户身份认证失败
500	服务器内部错误

## 错误码

请参见[错误码](#)。

### 4.1.9 列举所有资源

#### 功能介绍

返回当前用户下所有资源，需要当前用户有rms:resources:list权限。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/all-resources

表 4-60 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-61 Query 参数

参数	是否必选	参数类型	描述
region_id	否	String	区域ID。 最大长度：36
ep_id	否	String	企业项目ID。 最大长度：36

参数	是否必选	参数类型	描述
type	否	String	资源类型列表，资源类型的格式为provider.type。 最大长度：40
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：100
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
id	否	String	资源ID。 最大长度：512
name	否	String	资源名称。 最大长度：256
tags	否	Array	标签列表，标签值的格式为key或key=value。 数组长度：1 - 5

## 请求参数

表 4-62 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-63 响应 Body 参数

参数	参数类型	描述
resources	Array of <b>ResourceEntity</b> objects	资源列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-64 ResourceEntity

参数	参数类型	描述
id	String	资源ID。
name	String	资源名称。
provider	String	云服务名称。
type	String	资源类型。
region_id	String	区域ID。
project_id	String	Openstack中的项目ID。
project_name	String	Openstack中的项目名称。
ep_id	String	企业项目ID。
ep_name	String	企业项目名称。
checksum	String	资源详情校验码。
created	String	资源创建时间。
updated	String	资源更新时间。
provisioning_state	String	资源操作状态。枚举值：Succeeded（资源操作成功）、Failed（资源操作异常）、Canceled（资源操作取消）、Processing（资源操作正在进行中）。
tags	Map<String,String>	资源标签。
properties	Map<String,Object>	资源详细属性。

表 4-65 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-66 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-67 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-68 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

- 查询当前用户下所有资源。

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/all-resources
```

- 查询当前用户default企业项目下的资源，返回前100条。

GET https://{endpoint}/v1/resource-manager/domains/{domain\_id}/all-resources?limit=100&ep\_id=0

## 响应示例

**状态码： 200**

操作成功。

```
{
  "resources": [ {
    "id": "3ccd9191-6a5e-4939-a971-4652db18b370",
    "name": "elb-265a",
    "provider": "elb",
    "type": "loadbalancers",
    "region_id": "regionid1",
    "project_id": "05498e12458025102ff5c0061a584a9f",
    "project_name": "regionid1_region_service",
    "ep_id": "0",
    "ep_name": "default",
    "checksum": "6e0271b107b764b19acb235f45c0d852f72104fe1d4b32970686e7eae8e87bf4",
    "created": "2020-02-29T09:39:19Z",
    "updated": "2020-02-29T09:39:19Z",
    "provisioning_state": "Succeeded",
    "tags": { },
    "properties": {
      "tenant_id": "05498e12458025102ff5c0061a584a9f",
      "listeners": [ {
        "id": "37de3be0-1803-43e2-9bb5-243b4b30b771"
      } ],
      "provisioning_status": "ACTIVE",
      "description": ""
    }
  }, {
    "id": "a6e56d05501944d3b2507ba506a43744",
    "name": "console",
    "provider": "cdn",
    "type": "domains",
    "region_id": "global",
    "project_id": "",
    "project_name": "",
    "ep_id": "0",
    "ep_name": "default",
    "checksum": "56afa8b76428f90e9abf5cbf33535d8816166114d32eeb119658d6c59eceda",
    "created": "2020-01-04T13:42:37Z",
    "updated": "2020-01-15T04:23:01Z",
    "provisioning_state": "Succeeded",
    "tags": { },
    "properties": {
      "domain_name": "console",
      "domain_status": "offline",
      "business_type": "WEB",
      "modify_time": 1579062181463,
      "cname": "console"
    }
  } ],
  "page_info": {
    "current_count": 2,
    "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

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

public class ListAllResourcesSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAllResourcesRequest request = new ListAllResourcesRequest();
        request.withRegionId("<region_id>");
        request.withEpId("<ep_id>");
        request.withType("<type>");
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        request.withId("<id>");
        request.withName("<name>");
        request.withTags();
        try {
            ListAllResourcesResponse response = client.listAllResources(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *
```

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListAllResourcesRequest()
        request.region_id = "<region_id>"
        request.ep_id = "<ep_id>"
        request.type = "<type>"
        request.limit = <limit>
        request.marker = "<marker>"
        request.id = "<id>"
        request.name = "<name>"
        request.tags =
        response = client.list_all_resources(request)
        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/global"
    config "github.com/ HuaweiCloud/ HuaweiCloud-SDK-Go-v3/services/config/v1"
    "github.com/ HuaweiCloud/ HuaweiCloud-SDK-Go-v3/services/config/v1/model"
    region "github.com/ HuaweiCloud/ HuaweiCloud-SDK-Go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAllResourcesRequest{}
    regionIdRequest:= "<region_id>"
```



```
request.RegionId = &regionIdRequest
epIdRequest:= "<ep_id>"
request.EpId = &epIdRequest
typeRequest:= "<type>"
request.Type = &typeRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
idRequest:= "<id>"
request.Id = &idRequest
nameRequest:= "<name>"
request.Name = &nameRequest
response, err := client.ListAllResources(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.1.10 查询账号下的单个资源

#### 功能介绍

查询当前账号下的单个资源。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/all-resources/{resource\_id}

表 4-69 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
resource_id	是	String	资源ID。 最大长度：512

## 请求参数

表 4-70 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-71 响应 Body 参数

参数	参数类型	描述
id	String	资源ID。
name	String	资源名称。
provider	String	云服务名称。
type	String	资源类型。
region_id	String	区域ID。
project_id	String	Openstack中的项目ID。
project_name	String	Openstack中的项目名称。
ep_id	String	企业项目ID。
ep_name	String	企业项目名称。
checksum	String	资源详情校验码。
created	String	资源创建时间。
updated	String	资源更新时间。

参数	参数类型	描述
provisioning_state	String	资源操作状态。枚举值：Succeeded（资源操作成功）、Failed（资源操作异常）、Canceled（资源操作取消）、Processing（资源操作正在进行中）。
tags	Map<String,String>	资源标签。
properties	Map<String,Object>	资源详细属性。

状态码： 400

表 4-72 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-73 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-74 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
GET https://{endpoint}/v1/resource-manager/domains/059b5c937100d3e40ff0c00a7675a0a0/all-resources/00337e93-82d1-40ca-911f-07cff94587cc
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "id": "00337e93-82d1-40ca-911f-07cff94587cc",
  "name": "dev_machine",
  "provider": "ecs",
  "type": "cloudservers",
  "region_id": "regionid4",
  "project_id": "39c2af998c334ed6bcbb75b27318f7cc",
  "project_name": "project_name",
  "ep_id": "0",
  "ep_name": "chloe",
  "checksum": "3a0075409edb156a74e041b7479f0d5993be1d62b4ccd2af3a1dd01ec80c8b39",
  "created": "2019-11-20T06:24:43Z",
  "updated": "2020-07-17T08:30:52Z",
  "provisioning_state": "Succeeded",
  "tags": {
    "usage": "展示"
  },
  "properties": {
    "accessIPv4": "",
    "hostName": "dev-machine",
    "addresses": [ {
      "OsExtIpsType": "fixed",
      "OsExtIpsPortId": "f2fa750a-e2ab-434f-b14a-bfe7c8cea0cc",
      "addr": "192.168.1.212",
      "version": 4,
      "OsExtIpsMacAddr": "fa:16:3e:6e:cf:33"
    }, {
      "OsExtIpsType": "floating",
      "OsExtIpsPortId": "f2fa750a-e2ab-434f-b14a-bfe7c8cea0cc",
      "addr": "100.85.225.33",
      "version": 4,
      "OsExtIpsMacAddr": "fa:16:3e:6e:cf:33"
    } ],
    "accessIPv6": "",
    "metadata": {
      "chargingMode": "0",
      "meteringImageType": "private",
      "imageName": "resource-manager-devmachine-template",
      "meteringImageId": "9bcaace4-b8da-4008-a352-3f72e1f25333",
      "meteringResourcesPerCode": "si2.large.2.linux",
      "vpclId": "cf403ef5-90df-4e7e-829d-5d21b1cb7d1e",
      "osBit": "64"
    },
    "OsExtStsVmState": "active",
    "configDrive": "",
    "OsExtStsPowerState": 1,
    "hostId": "3c381dcfc3e628c1a504ad94ba8c4e89081306455273701333f32921",
    "securityGroup": [ {
      "name": "default",
      "id": "5d55b397-ad9c-462d-af72-6599cb941c49"
    } ],
    "ExtVolumesAttached": [ {
      "bootIndex": "0",
      "id": "010d940e-a73e-417b-85ae-51b76c0d2ba0",
      "device": "/dev/vda"
    } ],
    "userId": "e311190745e94cc09d62d5779e55912d",
    "flavor": {
      "disk": "0",
      "name": "Si2.large.2",
      "id": "Si2.large.2",
      "vcpus": "2",
      "ram": "4096"
    }
  }
}
```

```
    },
    "OsDcfDiskConfig" : "MANUAL",
    "hostStatus" : "UP",
    "OsSrvUsgLaunchedAt" : "2019-11-20T06:24:56.000000",
    "OsExtAz" : "regionid4a",
    "progress" : 0,
    "locked" : false,
    "OS-EXT-SRV-ATTR" : {
      "hostName" : "dev-machine",
      "kernelId" : "",
      "ramdiskId" : "",
      "reservationId" : "r-hhux9o7m",
      "instanceName" : "instance-0009cb50",
      "host" : "regionid4a-pod01.regionid4",
      "rootDeviceName" : "/dev/vda",
      "hypervisorHostName" : "nova001@2",
      "launchIndex" : 0
    },
    "status" : "ACTIVE"
  }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowResourceDetailSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowResourceDetailRequest request = new ShowResourceDetailRequest();
        try {
            ShowResourceDetailResponse response = client.showResourceDetail(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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowResourceDetailRequest()
        response = client.show_resource_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
```

```
client := config.NewConfigClient(  
    config.ConfigClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.ShowResourceDetailRequest{}  
response, err := client.ShowResourceDetail(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.1.11 列举资源标签

#### 功能介绍

查询当前账号下所有资源的标签。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/all-resources/tags

表 4-75 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-76 Query 参数

参数	是否必选	参数类型	描述
key	否	String	标签键名。 最大长度：128
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：100

## 请求参数

表 4-77 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200



表 4-78 响应 Body 参数

参数	参数类型	描述
tags	Array of <b>TagEntity</b> objects	标签列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-79 TagEntity

参数	参数类型	描述
key	String	标签键。
value	Array of strings	标签键对应的标签值列表。

表 4-80 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-81 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-82 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-83 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
GET https://{endpoint}/v1/resource-manager/domains/059b5c937100d3e40ff0c00a7675a0a0/all-resources/  
tags
```

## 响应示例

状态码： 200

操作成功。

```
{  
  "tags" : [ {  
    "key" : "chloe",  
    "value" : [ "a", "b" ]  
  } ],  
  "page_info" : {  
    "current_count" : 1,  
    "next_marker" : 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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;
```

```
public class ListAllTagsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAllTagsRequest request = new ListAllTagsRequest();
        request.withKey("<key>");
        request.withMarker("<marker>");
        request.withLimit(<limit>);
        try {
            ListAllTagsResponse response = client.listAllTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListAllTagsRequest()
        request.key = "<key>"
        request.marker = "<marker>"
```

```
request.limit = <limit>
response = client.list_all_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAllTagsRequest{}
    keyRequest:= "<key>"
    request.Key = &keyRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ListAllTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.1.12 查询资源数量

#### 功能介绍

查询当前账号的资源数量。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/all-resources/count

表 4-84 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-85 Query 参数

参数	是否必选	参数类型	描述
id	否	String	资源ID。 最大长度：512
name	否	String	资源名称。 最大长度：256

参数	是否必选	参数类型	描述
type	否	Array	资源类型列表，资源类型的格式为provider.type。 数组长度：1 - 100
region_id	否	Array	区域ID列表。 数组长度：1 - 10
ep_id	否	Array	企业项目ID列表。 数组长度：1 - 10
project_id	否	Array	项目ID。 数组长度：1 - 10
tags	否	Array	标签列表，标签值的格式为key或key=value。 数组长度：1 - 5

## 请求参数

表 4-86 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-87 响应 Body 参数

参数	参数类型	描述
total_count	Integer	资源总数。

状态码：400

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	错误消息内容。

状态码： 403

表 4-89 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-90 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
GET https://{endpoint}/v1/resource-manager/domains/059b5c937100d3e40ff0c00a7675a0a0/all-resources/count
```

## 响应示例

状态码： 200

操作成功。

```
{  
  "total_count": 345  
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

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

public class CountAllResourcesSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        CountAllResourcesRequest request = new CountAllResourcesRequest();
        request.withId("<id>");
        request.withName("<name>");
        request.withType();
        request.withRegionId();
        request.withEpId();
        request.withProjectId();
        request.withTags();
        try {
            CountAllResourcesResponse response = client.countAllResources(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```



```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = CountAllResourcesRequest()
    request.id = "<id>"
    request.name = "<name>"
    request.type =
    request.region_id =
    request.ep_id =
    request.project_id =
    request.tags =
    response = client.count_all_resources(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CountAllResourcesRequest{}
    idRequest := "<id>"
    request.Id = &idRequest
    nameRequest := "<name>"
    request.Name = &nameRequest
    response, err := client.CountAllResources(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
```

```
    fmt.Println(err)
  }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.1.13 列举资源概要

#### 功能介绍

查询当前账号的资源概览。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/all-resources/summary

表 4-91 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度： <b>36</b>

表 4-92 Query 参数

参数	是否必选	参数类型	描述
name	否	String	资源名称。 最大长度：256
type	否	Array	资源类型列表，资源类型的格式为provider.type。 数组长度：1 - 100
region_id	否	Array	区域ID列表。 数组长度：1 - 10
ep_id	否	Array	企业项目ID列表。 数组长度：1 - 10
project_id	否	Array	项目ID。 数组长度：1 - 10
tags	否	Array	标签列表，标签值的格式为key或key=value。 数组长度：1 - 5

## 请求参数

表 4-93 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-94 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of <a href="#">ResourceSummaryResponseItem</a> objects	资源概要信息列表。

表 4-95 ResourceSummaryResponseItem

参数	参数类型	描述
provider	String	云服务名称。
types	Array of <b>types</b> objects	资源类型列表。

表 4-96 types

参数	参数类型	描述
type	String	资源类型名称。
regions	Array of <b>regions</b> objects	区域列表。

表 4-97 regions

参数	参数类型	描述
region_id	String	区域ID。
count	Long	该资源类型在当前区域的数量。

状态码： 400

表 4-98 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-99 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-100 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
GET https://{endpoint}/v1/resource-manager/domains/059b5c937100d3e40ff0c00a7675a0a0/all-resources/summary
```

## 响应示例

状态码： 200

操作成功。

```
[ {  
  "provider": "obs",  
  "types": [ {  
    "type": "buckets",  
    "regions": [ {  
      "region_id": "regionid1",  
      "count": 5  
    } ]  
  } ]  
}, {  
  "provider": "ecs",  
  "types": [ {  
    "type": "cloudservers",  
    "regions": [ {  
      "region_id": "regionid2",  
      "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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CollectAllResourcesSummarySolution {
```

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

    ICredential auth = new GlobalCredentials()
        .withAk(ak)
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    CollectAllResourcesSummaryRequest request = new CollectAllResourcesSummaryRequest();
    request.withName("<name>");
    request.withType();
    request.withRegionId();
    request.withEpId();
    request.withProjectId();
    request.withTags();
    try {
        CollectAllResourcesSummaryResponse response = client.collectAllResourcesSummary(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CollectAllResourcesSummaryRequest()
```

```
request.name = "<name>"
request.type =
request.region_id =
request.ep_id =
request.project_id =
request.tags =
response = client.collect_all_resources_summary(request)
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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CollectAllResourcesSummaryRequest{}
    nameRequest := "<name>"
    request.Name = &nameRequest
    response, err := client.CollectAllResourcesSummary(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.2 资源记录器

### 4.2.1 查询记录器

#### 功能介绍

查询资源记录器的详细信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/tracker-config

表 4-101 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

#### 请求参数

表 4-102 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。



参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 200

表 4-103 响应 Body 参数

参数	参数类型	描述
channel	<a href="#">ChannelConfigBody</a> object	tracker通道配置。
selector	<a href="#">SelectorConfigBody</a> object	tracker选择器。
agency_name	String	IAM委托名称。

表 4-104 ChannelConfigBody

参数	参数类型	描述
smn	<a href="#">TrackerSMNChannelConfigBody</a> object	SMN通道设置对象。跨账号授予SMN主题发送通知的权限请参考《用户指南- 资源记录器- 开启/配置/修改资源记录器》中的“跨账号授权”内容。
obs	<a href="#">TrackerOBSCChannelConfigBody</a> object	OBS设置对象。跨账号授予OBS桶转储文件的权限请参考《用户指南- 资源记录器- 开启/配置/修改资源记录器》中的“跨账号授权”内容。

表 4-105 TrackerSMNChannelConfigBody

参数	参数类型	描述
region_id	String	区域id。
project_id	String	项目id。
topic_urn	String	SMN主题urn。

表 4-106 TrackerOBSChannelConfigBody

参数	参数类型	描述
bucket_name	String	OBS桶名称。 最大长度：63
bucket_prefix	String	OBS桶前缀。 最大长度：256
region_id	String	区域id。

表 4-107 SelectorConfigBody

参数	参数类型	描述
all_supported	Boolean	是否选择所有支持的资源。
resource_types	Array of strings	资源类型列表。

状态码：400

表 4-108 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-109 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-110 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-111 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "agency_name" : "rms_tracker_agency",
  "channel" : {
    "obs" : {
      "bucket_name" : "rmttest",
      "region_id" : "regionid1"
    },
    "smn" : {
      "project_id" : "059b5e0a2500d5552fa1c00adada8c06",
      "region_id" : "regionid1",
      "topic_urn" : "urn:smn:regionid1:059b5e0a2500d5552fa1c00adada8c06:agencytest"
    }
  },
  "selector" : {
    "all_supported" : true,
    "resource_types" : [ ]
  }
}
```

## 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.GlobalCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowTrackerConfigSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTrackerConfigRequest request = new ShowTrackerConfigRequest();
        try {
            ShowTrackerConfigResponse response = client.showTrackerConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

```
try:
    request = ShowTrackerConfigRequest()
    response = client.show_tracker_config(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。

状态码	描述
403	用户认证失败，或没有操作权限。
404	没有找到资源记录器。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.2.2 删除记录器

### 功能介绍

删除资源记录器。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/resource-manager/domains/{domain\_id}/tracker-config

表 4-112 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

### 请求参数

表 4-113 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 400

表 4-114 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-115 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-116 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-117 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeleteTrackerConfigSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteTrackerConfigRequest request = new DeleteTrackerConfigRequest();
        try {
            DeleteTrackerConfigResponse response = client.deleteTrackerConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *
```



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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeleteTrackerConfigRequest()
        response = client.delete_tracker_config(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败，或没有操作权限。
404	未找到资源记录器。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.2.3 创建或更新记录器

### 功能介绍

创建或更新资源记录器，只能存在一个资源记录器。

### 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1/resource-manager/domains/{domain\_id}/tracker-config

表 4-118 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

## 请求参数

表 4-119 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-120 请求 Body 参数

参数	是否必选	参数类型	描述
channel	是	<a href="#">ChannelConfigBody</a> object	tracker通道配置。
selector	是	<a href="#">SelectorConfigBody</a> object	tracker选择器。
agency_name	是	String	IAM委托名称。

表 4-121 ChannelConfigBody

参数	是否必选	参数类型	描述
smn	否	<a href="#">TrackerSMNChannelConfigBody</a> object	SMN通道设置对象。跨账号授予SMN主题发送通知的权限请参考《用户指南- 资源记录器- 开启/配置/修改资源记录器》中的“跨账号授权”内容。
obs	否	<a href="#">TrackerOBSCChannelConfigBody</a> object	OBS设置对象。跨账号授予OBS桶转储文件的权限请参考《用户指南- 资源记录器- 开启/配置/修改资源记录器》中的“跨账号授权”内容。

表 4-122 TrackerSMNChannelConfigBody

参数	是否必选	参数类型	描述
region_id	是	String	区域id。
project_id	是	String	项目id。
topic_urn	是	String	SMN主题urn。

表 4-123 TrackerOBSChannelConfigBody

参数	是否必选	参数类型	描述
bucket_name	是	String	OBS桶名称。 最大长度： <b>63</b>
bucket_prefix	否	String	OBS桶前缀。 最大长度： <b>256</b>
region_id	是	String	区域id。

表 4-124 SelectorConfigBody

参数	是否必选	参数类型	描述
all_supported	是	Boolean	是否选择所有支持的资源。
resource_types	是	Array of strings	资源类型列表。

## 响应参数

状态码：400

表 4-125 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-126 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-127 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-128 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

创建资源记录器，并开启消息推送与内容存储。

```
PUT /v1/resource-manager/domains/{domain_id}/tracker-config
```

```
{
  "channel": {
    "smn": {
      "region_id": "regionid1",
      "project_id": "39c2af998c334ed6bcbb75b27318f7b5",
      "topic_urn": "urn:smn:regionid1:39c2af998c334ed6bcbb75b27318f7b5:resource-manager-test"
    },
    "obs": {
      "bucket_name": "config-snapshot",
      "region_id": "regionid1"
    }
  },
  "selector": {
    "all_supported": true,
    "resource_types": []
  },
  "agency_name": "rms_tracker_agency"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

创建资源记录器，并开启消息推送与内容存储。

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

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

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

public class CreateTrackerConfigSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateTrackerConfigRequest request = new CreateTrackerConfigRequest();
        TrackerConfigBody body = new TrackerConfigBody();
        SelectorConfigBody selectorbody = new SelectorConfigBody();
        selectorbody.withAllSupported(true);
        TrackerOBSChannelConfigBody obsChannel = new TrackerOBSChannelConfigBody();
        obsChannel.withBucketName("config-snapshot")
            .withRegionId("regionid1");
        TrackerSMNChannelConfigBody smnChannel = new TrackerSMNChannelConfigBody();
        smnChannel.withRegionId("regionid1")
            .withProjectId("39c2af998c334ed6bcbb75b27318f7b5")
            .withTopicUrn("urn:smn:regionid1:39c2af998c334ed6bcbb75b27318f7b5:resource-manager-test");
        ChannelConfigBody channelbody = new ChannelConfigBody();
        channelbody.withSmn(smnChannel)
            .withObs(obsChannel);
        body.withAgencyName("rms_tracker_agency");
        body.withSelector(selectorbody);
        body.withChannel(channelbody);
        request.withBody(body);
        try {
            CreateTrackerConfigResponse response = client.createTrackerConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

创建资源记录器，并开启消息推送与内容存储。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CreateTrackerConfigRequest()
        selectorbody = SelectorConfigBody(
            all_supported=True
        )
        obsChannel = TrackerOBSChannelConfigBody(
            bucket_name="config-snapshot",
            region_id="regionid1"
        )
        smnChannel = TrackerSMNChannelConfigBody(
            region_id="regionid1",
            project_id="39c2af998c334ed6bcb75b27318f7b5",
            topic_urn="urn:smn:regionid1:39c2af998c334ed6bcb75b27318f7b5:resource-manager-test"
        )
        channelbody = ChannelConfigBody(
            smn=smnChannel,
            obs=obsChannel
        )
        request.body = TrackerConfigBody(
            agency_name="rms_tracker_agency",
            selector=selectorbody,
            channel=channelbody
        )
        response = client.create_tracker_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

创建资源记录器，并开启消息推送与内容存储。

```
package main

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

func main() {
```

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

auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateTrackerConfigRequest{}
selectorbody := &model.SelectorConfigBody{
    AllSupported: true,
}
obsChannel := &model.TrackerObsChannelConfigBody{
    BucketName: "config-snapshot",
    RegionId: "regionid1",
}
smnChannel := &model.TrackerSmnChannelConfigBody{
    RegionId: "regionid1",
    ProjectId: "39c2af998c334ed6bcb75b27318f7b5",
    TopicUrn: "urn:smn:regionid1:39c2af998c334ed6bcb75b27318f7b5:resource-manager-test",
}
channelbody := &model.ChannelConfigBody{
    Smn: smnChannel,
    Obs: obsChannel,
}
request.Body = &model.TrackerConfigBody{
    AgencyName: "rms_tracker_agency",
    Selector: selectorbody,
    Channel: channelbody,
}
response, err := client.CreateTrackerConfig(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败，或没有操作权限。
404	资源未找到。



状态码	描述
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.3 资源关系

### 4.3.1 列举资源关系

#### 功能介绍

指定资源ID，查询该资源与其他资源的关联关系，可以指定关系方向为"in" 或者 "out"。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/resources/{resource\_id}/relations

表 4-129 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
resource_id	是	String	资源ID。 最大长度：512

表 4-130 Query 参数

参数	是否必选	参数类型	描述
direction	是	String	资源关系的指向。
limit	否	Integer	最大的返回数量。 最小值：1 最大值：1000

参数	是否必选	参数类型	描述
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-131 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-132 响应 Body 参数

参数	参数类型	描述
relations	Array of <a href="#">ResourceRelation</a> objects	资源关系列表。
page_info	<a href="#">PageInfo</a> object	分页对象。

表 4-133 ResourceRelation

参数	参数类型	描述
relation_type	String	关系类型。
from_resource_type	String	源资源类型。
to_resource_type	String	目的资源类型。

参数	参数类型	描述
from_resource_id	String	源资源ID。
to_resource_id	String	目的资源ID。

表 4-134 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-135 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-136 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-137 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-138 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/resources/{resource_id}/relations?direction=out&limit=1000
```

## 响应示例

状态码： 200

操作成功。

```
{
  "relations": [ {
    "relation_type": "isAttachedTo",
    "from_resource_type": "ecs.cloudservers",
    "to_resource_type": "evs.volumes",
    "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
    "to_resource_id": "0075ed19-59dd-49be-961d-117bb6bfd3e"
  }, {
    "relation_type": "contains",
    "from_resource_type": "ecs.cloudservers",
    "to_resource_type": "vpc.publicips",
    "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
    "to_resource_id": "3813d6d3-ef88-47b1-b343-cdf6390c6dcb"
  }, {
    "relation_type": "isAssociatedWith",
    "from_resource_type": "ecs.cloudservers",
    "to_resource_type": "vpc.securityGroups",
    "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
    "to_resource_id": "8cca3002-00af-4812-a853-b7a6fbee06a4"
  }, {
    "relation_type": "isAttachedTo",
    "from_resource_type": "ecs.cloudservers",
    "to_resource_type": "evs.volumes",
    "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
    "to_resource_id": "f4a107eb-4c6d-4dc8-88d8-de337923956f"
  }, {
    "relation_type": "isContainedIn",
    "from_resource_type": "ecs.cloudservers",
    "to_resource_type": "vpc.vpcs",
    "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
```

```
"to_resource_id" : "ff13d70d-17e5-4ec8-945a-c874e0db99d3"
}],
"page_info" : {
  "current_count" : 5,
  "next_marker" : 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowResourceRelationsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowResourceRelationsRequest request = new ShowResourceRelationsRequest();
        request.withDirection(ShowResourceRelationsRequest.DirectionEnum.fromValue("<direction>"));
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        try {
            ShowResourceRelationsResponse response = client.showResourceRelations(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowResourceRelationsRequest()
        request.direction = "<direction>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.show_resource_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.NewConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.ShowResourceRelationsRequest{}
request.Direction = model.GetShowResourceRelationsRequestDirectionEnum().<DIRECTION>
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
response, err := client.ShowResourceRelations(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.3.2 列举资源关系详情

#### 功能介绍

指定资源ID，查询该资源与其他资源的关联关系，可以指定关系方向为“in”或者“out”，需要当账号有rms:resources:getRelation权限。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/all-resources/{resource\_id}/relations

表 4-139 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
resource_id	是	String	资源ID。 最大长度：512

表 4-140 Query 参数

参数	是否必选	参数类型	描述
direction	是	String	资源关系的指向。
limit	否	Integer	最大的返回数量。 最小值：1 最大值：1000
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-141 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200



表 4-142 响应 Body 参数

参数	参数类型	描述
relations	Array of <b>ResourceRelation</b> objects	资源关系列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-143 ResourceRelation

参数	参数类型	描述
relation_type	String	关系类型。
from_resource_type	String	源资源类型。
to_resource_type	String	目的资源类型。
from_resource_id	String	源资源ID。
to_resource_id	String	目的资源ID。

表 4-144 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-145 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-146 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-147 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-148 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
GET https://{endpoint}/v1/resource-manager/domains/{domain_id}/all-resources/{resource_id}/relations?direction=out
```

## 响应示例

状态码： 200

操作成功。

```
{
  "relations": [ {
    "relation_type": "isAttachedTo",
    "from_resource_type": "ecs.cloudservers",
    "to_resource_type": "evs.volumes",
    "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
    "to_resource_id": "0075ed19-59dd-49be-961d-117bb6bfd3e"
  }, {
    "relation_type": "contains",
    "from_resource_type": "ecs.cloudservers",
    "to_resource_type": "vpc.publicips",
    "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
```

```
"to_resource_id" : "3813d6d3-ef88-47b1-b343-cdf6390c6dcb"
}, {
  "relation_type" : "isAssociatedWith",
  "from_resource_type" : "ecs.cloudservers",
  "to_resource_type" : "vpc.securityGroups",
  "from_resource_id" : "6af96128-d58d-426c-91e0-b38144c0f112",
  "to_resource_id" : "8cca3002-00af-4812-a853-b7a6fbee06a4"
}, {
  "relation_type" : "isAttachedTo",
  "from_resource_type" : "ecs.cloudservers",
  "to_resource_type" : "evs.volumes",
  "from_resource_id" : "6af96128-d58d-426c-91e0-b38144c0f112",
  "to_resource_id" : "f4a107eb-4c6d-4dc8-88d8-de337923956f"
}, {
  "relation_type" : "isContainedIn",
  "from_resource_type" : "ecs.cloudservers",
  "to_resource_type" : "vpc.vpcs",
  "from_resource_id" : "6af96128-d58d-426c-91e0-b38144c0f112",
  "to_resource_id" : "ff13d70d-17e5-4ec8-945a-c874e0db99d3"
}],
"page_info" : {
  "current_count" : 5,
  "next_marker" : 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowResourceRelationsDetailSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

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

        ShowResourceRelationsDetailRequest request = new ShowResourceRelationsDetailRequest();
        request.withDirection(ShowResourceRelationsDetailRequest.DirectionEnum.fromValue("<direction>"));
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        try {
```

```
        ShowResourceRelationsDetailResponse response = client.showResourceRelationsDetail(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowResourceRelationsDetailRequest()
        request.direction = "<direction>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.show_resource_relations_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

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

auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowResourceRelationsDetailRequest{}
request.Direction = model.GetShowResourceRelationsDetailRequestDirectionEnum().<DIRECTION>
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
response, err := client.ShowResourceRelationsDetail(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.4 资源历史

## 4.4.1 查询资源历史

### 功能介绍

查询资源与资源关系的变更历史。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/resources/{resource\_id}/history

表 4-149 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
resource_id	是	String	资源ID。 最大长度：512

表 4-150 Query 参数

参数	是否必选	参数类型	描述
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
earlier_time	否	Long	指定查询范围的起始时间点，如果不设置此参数，默认为最早的时间。
later_time	否	Long	指定查询范围的结束时间点，如果不设置此参数，默认为当前时间。
chronological_order	否	String	指定返回数据的时间顺序，默认为倒序。

## 请求参数

表 4-151 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 200

表 4-152 响应 Body 参数

参数	参数类型	描述
items	Array of <a href="#">HistoryItem</a> objects	资源历史列表。
page_info	<a href="#">PageInfo</a> object	分页对象。

表 4-153 HistoryItem

参数	参数类型	描述
domain_id	String	用户id。
resource_id	String	资源id。
resource_type	String	资源类型。
capture_time	String	该资源系统捕获时间。
status	String	资源状态。
relations	Array of <a href="#">ResourceRelation</a> objects	资源关系列表。
resource	<a href="#">ResourceEntity</a> object	资源对象。

表 4-154 ResourceRelation

参数	参数类型	描述
relation_type	String	关系类型。
from_resource_type	String	源资源类型。
to_resource_type	String	目的资源类型。
from_resource_id	String	源资源ID。
to_resource_id	String	目的资源ID。

表 4-155 ResourceEntity

参数	参数类型	描述
id	String	资源ID。
name	String	资源名称。
provider	String	云服务名称。
type	String	资源类型。
region_id	String	区域ID。
project_id	String	Openstack中的项目ID。
project_name	String	Openstack中的项目名称。
ep_id	String	企业项目ID。
ep_name	String	企业项目名称。
checksum	String	资源详情校验码。
created	String	资源创建时间。
updated	String	资源更新时间。
provisioning_state	String	资源操作状态。枚举值：Succeeded（资源操作成功）、Failed（资源操作异常）、Canceled（资源操作取消）、Processing（资源操作正在进行中）。
tags	Map<String,String>	资源标签。
properties	Map<String,Object>	资源详细属性。



表 4-156 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码：400

表 4-157 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-158 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-159 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
GET /v1/resource-manager/domains/{domain_id}/resources/{resource_id}/history?  
earlier_time=1595865600000&later_time=1603875761000&limit=10
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "items": [ {
    "domain_id": "9548dff51b6849fab3d7353d631158ee",
    "resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
    "resource_type": "ecs.cloudservers",
    "capture_time": "2020-10-12T07:44:40.167Z",
    "status": "ResourceChanged.UPDATE",
    "relations": [ {
      "relation_type": "contains",
      "from_resource_type": "ecs.cloudservers",
      "to_resource_type": "vpc.publicips",
      "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
      "to_resource_id": "3813d6d3-ef88-47b1-b343-cdf6390c6dcb"
    }, {
      "relation_type": "isAttachedTo",
      "from_resource_type": "ecs.cloudservers",
      "to_resource_type": "evs.volumes",
      "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
      "to_resource_id": "0075ed19-59dd-49be-961d-117bb6fbfd3e"
    }, {
      "relation_type": "isAssociatedWith",
      "from_resource_type": "ecs.cloudservers",
      "to_resource_type": "vpc.securityGroups",
      "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
      "to_resource_id": "8cca3002-00af-4812-a853-b7a6fbee06a4"
    }, {
      "relation_type": "isAttachedTo",
      "from_resource_type": "ecs.cloudservers",
      "to_resource_type": "evs.volumes",
      "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
      "to_resource_id": "f4a107eb-4c6d-4dc8-88d8-de337923956f"
    }, {
      "relation_type": "isContainedIn",
      "from_resource_type": "ecs.cloudservers",
      "to_resource_type": "vpc.vpcs",
      "from_resource_id": "6af96128-d58d-426c-91e0-b38144c0f112",
      "to_resource_id": "ff13d70d-17e5-4ec8-945a-c874e0db99d3"
    }
  ]
}, {
  "resource": {
    "id": "6af96128-d58d-426c-91e0-b38144c0f112",
    "name": "david-test-cce-49831-bh9ax",
    "provider": "ecs",
    "type": "cloudservers",
    "region_id": "regionid1",
    "project_id": "15caceb7599f4ee3a770dbb70ad69b2b",
    "project_name": "project_name",
    "ep_id": "0",
    "ep_name": "default",
    "checksum": "30482e7e4f2fc6e9842689c5d498cd13862def3c7369e9216af87c2198557b28",
    "created": "2020-10-12T07:44:10Z",
    "updated": "2020-10-12T07:44:38Z",
    "provisioning_state": "Succeeded",
    "tags": {
      "CCE-Dynamic-Provisioning-Node": "80fbd7d5-0c5e-11eb-bc0e-0255ac101e85"
    }
  },
  "properties": {
    "accessipv4": "",
    "hostName": "david-test-cce-49831-bh9ax",
    "addresses": [ {
      "OsExtIpsType": "fixed",
      "OsExtIpsPortId": "a7bfdd6f-3a56-4bbb-b37e-1b9ccac1ac96",
      "addr": "192.168.238.230",
      "version": 4,
    }
  ]
}
]
```

```
    "OsExtIpsMacAddr" : "fa:16:3e:99:ec:b1"
  },
  "accessIpsV6" : "",
  "metadata" : {
    "chargingMode" : "0",
    "meteringImageType" : "gold",
    "imageName" : "CCE_images_EulerOS-Node-CCE20.7.1.B003-v1.17.9-r0-amd64-
f8389e5c3d65dc7b5bf8f10924d2fc16e35588fc-20200808144721",
    "meteringImageId" : "e8b32b27-a103-4a1f-b955-5d6be46ae9da",
    "meteringResourcesPerCode" : "si3.large.2.linux",
    "vpclId" : "ff13d70d-17e5-4ec8-945a-c874e0db99d3",
    "osBit" : "64"
  },
  "OsExtStsVmState" : "active",
  "configDrive" : "",
  "OsExtStsPowerState" : 1,
  "hostId" : "7f007da24f79bc37782d3440e3d4faa374c0eb555c1dfc98b687c996",
  "securityGroup" : [ {
    "name" : "david-test-cce-cce-node-8x31o",
    "id" : "8cca3002-00af-4812-a853-b7a6fbee06a4"
  } ],
  "ExtVolumesAttached" : [ {
    "bootIndex" : "0",
    "id" : "f4a107eb-4c6d-4dc8-88d8-de337923956f",
    "device" : "/dev/vda"
  }, {
    "bootIndex" : "-1",
    "id" : "0075ed19-59dd-49be-961d-117bb6bfd3e",
    "device" : "/dev/vdb"
  } ],
  "userId" : "07729a673f8026624f3ec006f3b3cda9",
  "flavor" : {
    "disk" : "0",
    "name" : "Si3.large.2",
    "id" : "Si3.large.2",
    "vcpus" : "2",
    "ram" : "4096"
  },
  "OsDcfDiskConfig" : "MANUAL",
  "hostStatus" : "UP",
  "OsSrvUsgLaunchedAt" : "2020-10-12T07:44:23.000000",
  "OsExtAz" : "regionid1a",
  "progress" : 0,
  "locked" : false,
  "OS-EXT-SRV-ATTR" : {
    "hostName" : "david-test-cce-49831-bh9ax",
    "kernelId" : "",
    "ramdiskId" : "",
    "reservationId" : "r-l8y8evjo",
    "instanceName" : "instance-000b2c2d",
    "host" : "regionid1a-pod01.regionid1",
    "rootDeviceName" : "/dev/vda",
    "hypervisorHostName" : "nova001@83",
    "launchIndex" : 0
  },
  "status" : "ACTIVE"
}
} ],
"page_info" : {
  "current_count" : 1,
  "next_marker" :
"CAESJjE2MDI0ODg2Nzc1NDYtNWY4NDNA5NjU4YmI5MDUxZDBhMWEzZmMjGgQCXWaw"
}
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowResourceHistorySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowResourceHistoryRequest request = new ShowResourceHistoryRequest();
        request.withEarlierTime(<earlier_time>L);
        request.withLaterTime(<later_time>L);

        request.withChronologicalOrder(ShowResourceHistoryRequest.ChronologicalOrderEnum.fromValue("<chrono
logical_order>"));
        request.withMarker("<marker>");
        request.withLimit(<limit>);
        try {
            ShowResourceHistoryResponse response = client.showResourceHistory(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = ShowResourceHistoryRequest()
    request.earlier_time = <earlier_time>
    request.later_time = <later_time>
    request.chronological_order = "<chronological_order>"
    request.marker = "<marker>"
    request.limit = <limit>
    response = client.show_resource_history(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowResourceHistoryRequest{
        earlierTimeRequest:= int64(<earlier_time>)
        request.EarlierTime = &earlierTimeRequest
        laterTimeRequest:= int64(<later_time>)
        request.LaterTime = &laterTimeRequest
        chronologicalOrderRequest:=
        model.GetShowResourceHistoryRequestChronologicalOrderEnum().<CHRONOLOGICAL_ORDER>
        request.ChronologicalOrder = &chronologicalOrderRequest
```

```
markerRequest:= "<marker>"
request.Marker = &markerRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
response, err := client.ShowResourceHistory(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5 合规性

### 4.5.1 列出内置策略

#### 功能介绍

列出用户的内置策略。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/policy-definitions

## 请求参数

表 4-160 请求 Header 参数

参数	是否必选	参数类型	描述
X-Language	否	String	选择接口返回的信息的语言，默认为"zh-cn"中文，可选值：zh-cn和en-us。 缺省值： <b>zh-cn</b>
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 200

表 4-161 响应 Body 参数

参数	参数类型	描述
value	Array of <b>PolicyDefinition</b> objects	策略定义列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-162 PolicyDefinition

参数	参数类型	描述
id	String	策略id。
name	String	策略名称。
display_name	String	策略展示名。
policy_type	String	策略类型。
description	String	策略描述。
policy_rule_type	String	策略语法类型。
policy_rule	Object	策略规则。

参数	参数类型	描述
trigger_type	String	触发器类型，可选值："resource"和"period"。
keywords	Array of strings	关键词列表。
default_resource_types	Array of <b>default_resource_types</b> objects	默认资源类型列表
parameters	Map<String, <b>PolicyParameterDefinition</b> >	策略参数。

表 4-163 default\_resource\_types

参数	参数类型	描述
provider	String	云服务名称
type	String	资源类型

表 4-164 PolicyParameterDefinition

参数	参数类型	描述
name	String	策略参数名称。
description	String	策略参数描述。
allowed_values	Array of objects	策略参数允许值列表。
default_value	String	策略参数默认值。
minimum	Float	策略参数的最小值，当参数类型为Integer或Float时生效。
maximum	Float	策略参数的最大值，当参数类型为Integer或Float时生效。
min_items	Integer	策略参数的最小项数，当参数类型为Array时生效。
max_items	Integer	策略参数的最大项数，当参数类型为Array时生效。
min_length	Integer	策略参数的最小字符串长度或每项的最小字符串长度，当参数类型为String或Array时生效。



参数	参数类型	描述
max_length	Integer	策略参数的最大字符串长度或每项的最大字符串长度，当参数类型为String或Array时生效。
pattern	String	策略参数的字符串正则要求或每项的字符串正则要求，当参数类型为String或Array时生效。
type	String	策略参数类型。

表 4-165 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码：400

表 4-166 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-167 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-168 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-169 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [ {
    "id": "5fa365476eed194ccb2c04d1",
    "name": "volumes-encrypted-check",
    "policy_type": "builtin",
    "description": "已挂载的云硬盘未进行加密，视为“不合规”。",
    "policy_rule_type": "dsl",
    "policy_rule": {
      "allOf": [ {
        "value": "${resource().provider}",
        "comparator": "equals",
        "pattern": "evs"
      }, {
        "value": "${resource().type}",
        "comparator": "equals",
        "pattern": "volumes"
      }, {
        "value": "${resource().properties.status}",
        "comparator": "equals",
        "pattern": "in-use"
      }, {
        "anyOf": [ {
          "value": "${resource().properties.metadata}",
          "comparator": "notContainsKey",
          "pattern": "systemEncrypted"
        }, {
          "value": "${resource().properties.metadata.systemEncrypted}",
          "comparator": "equals",
          "pattern": "0"
        }
      ]
    }
  }
}]
}
```

```
    },
    "keywords" : [ "evs", "ecs" ],
    "parameters" : { }
  }, {
    "id" : "5fa9f89b6eed194ccb2c04db",
    "name" : "required-tag-check",
    "policy_type" : "builtin",
    "description" : "指定一个标签，不具有此标签的资源，视为“不合规”。",
    "policy_rule_type" : "dsl",
    "policy_rule" : {
      "anyOf" : [ {
        "value" : "${contains(resource().tags, parameters('specifiedTagKey'))}",
        "comparator" : "equals",
        "pattern" : false
      }, {
        "allOf" : [ {
          "value" : "${length(parameters('specifiedTagValue'))}",
          "comparator" : "greater",
          "pattern" : 0
        }, {
          "value" : "${getValue(resource().tags, parameters('specifiedTagKey'))}",
          "comparator" : "notIn",
          "pattern" : "${parameters('specifiedTagValue')}"
        }
      ]
    }
  ]
  },
  "keywords" : [ "tag" ],
  "parameters" : {
    "specifiedTagKey" : {
      "name" : null,
      "description" : "the specified tag key。",
      "allowed_values" : null,
      "default_value" : null,
      "type" : "String"
    },
    "specifiedTagValue" : {
      "name" : null,
      "description" : "The list of allowed tag value, permit all if empty。",
      "allowed_values" : null,
      "default_value" : null,
      "type" : "Array"
    }
  }
}, {
  "id" : "5fa265c0aa1e6afc05a0ff07",
  "name" : "allowed-images-by-id",
  "policy_type" : "builtin",
  "description" : "指定允许的镜像ID列表，ECS实例的镜像ID不在指定的范围内，视为“不合规”。",
  "policy_rule_type" : "dsl",
  "policy_rule" : {
    "allOf" : [ {
      "value" : "${resource().provider}",
      "comparator" : "equals",
      "pattern" : "ecs"
    }, {
      "value" : "${resource().type}",
      "comparator" : "equals",
      "pattern" : "cloudservers"
    }, {
      "value" : "${resource().properties.metadata.meteringImageId}",
      "comparator" : "notIn",
      "pattern" : "${parameters('listOfAllowedImages')}"
    }
  ]
  },
  "keywords" : [ "ecs", "ims" ],
  "parameters" : {
    "listOfAllowedImages" : {
      "name" : null,
      "description" : "The list of allowed image IDs。",

```

```
"allowed_values" : null,
"default_value" : null,
"type" : "Array"
}
}, {
  "id" : "5fa36558aa1e6afc05a3d0a7",
  "name" : "volume-inuse-check",
  "policy_type" : "builtin",
  "description" : "云硬盘未挂载给任何云服务器，视为“不合规”。",
  "policy_rule_type" : "dsl",
  "policy_rule" : {
    "allOf" : [ {
      "value" : "${resource().provider}",
      "comparator" : "equals",
      "pattern" : "evs"
    }, {
      "value" : "${resource().type}",
      "comparator" : "equals",
      "pattern" : "volumes"
    }, {
      "value" : "${resource().properties.status}",
      "comparator" : "equals",
      "pattern" : "available"
    }
  ]
}, {
  "keywords" : [ "evs" ],
  "parameters" : { }
}, {
  "id" : "5fa3a1196eed194ccb2c04d7",
  "name" : "allowed_volume_specs",
  "policy_type" : "builtin",
  "description" : "指定允许的云硬盘类型列表，云硬盘的类型不在指定的范围内，视为“不合规”。",
  "policy_rule_type" : "dsl",
  "policy_rule" : {
    "allOf" : [ {
      "value" : "${resource().provider}",
      "comparator" : "equals",
      "pattern" : "evs"
    }, {
      "value" : "${resource().type}",
      "comparator" : "equals",
      "pattern" : "volumes"
    }, {
      "value" : "${resource().properties.volumeType}",
      "comparator" : "notIn",
      "pattern" : "${parameters('listOfAllowedSpecs')}"
    }
  ]
}, {
  "keywords" : [ "evs" ],
  "parameters" : {
    "listOfAllowedSpecs" : {
      "name" : null,
      "description" : "The list of allowed disk specifications。",
      "allowed_values" : null,
      "default_value" : null,
      "type" : "Array"
    }
  }
}, {
  "id" : "5f8d549bffeccc14f1fb522a",
  "name" : "allowed_ecs_flavors",
  "policy_type" : "builtin",
  "description" : "ECS资源的规格不在指定的范围内，视为“不合规”。",
  "policy_rule_type" : "dsl",
  "policy_rule" : {
    "allOf" : [ {
      "value" : "${resource().provider}",
      "comparator" : "equals",
```

```
"pattern": "ecs"
}, {
  "value": "${resource().type}",
  "comparator": "equals",
  "pattern": "cloudservers"
}, {
  "value": "${resource().properties.flavor.name}",
  "comparator": "notIn",
  "pattern": "${parameters('listOfAllowedFlavors')}"
}]
},
"keywords": [ "ecs" ],
"parameters": {
  "listOfAllowedFlavors": {
    "name": null,
    "description": "The list of allowed flavor types。",
    "allowed_values": null,
    "default_value": null,
    "type": "Array"
  }
}
}, {
  "id": "5fa3a1196eed194ccb2c04d5",
  "name": "eip-unbound-check",
  "policy_type": "builtin",
  "description": "弹性公网IP未进行任何绑定, 视为“不合规”。",
  "policy_rule_type": "dsl",
  "policy_rule": {
    "allOf": [ {
      "value": "${resource().provider}",
      "comparator": "equals",
      "pattern": "vpc"
    }, {
      "value": "${resource().type}",
      "comparator": "equals",
      "pattern": "publicips"
    }, {
      "value": "${resource().properties.status}",
      "comparator": "equals",
      "pattern": "DOWN"
    }
  ]
},
"keywords": [ "vpc" ],
"parameters": { }
}, {
  "id": "5f8d5428ffecc14f1fb5205",
  "name": "ecs-instance-no-public-ip",
  "policy_type": "builtin",
  "description": "ECS资源具有公网IP, 视为“不合规”。",
  "policy_rule_type": "dsl",
  "policy_rule": {
    "allOf": [ {
      "value": "${resource().provider}",
      "comparator": "equals",
      "pattern": "ecs"
    }, {
      "value": "${resource().type}",
      "comparator": "equals",
      "pattern": "cloudservers"
    }, {
      "value": "${length(filter(resource().properties.addresses, lambda x: equals(x.OsExtIpsType,
floating)))}",
      "comparator": "greater",
      "pattern": 0
    }
  ]
},
"keywords": [ "ecs" ],
"parameters": { }
}, {
```

```
"id": "5fa265c0aa1e6afc05a0ff10",
"name": "ecs-instance-in-vpc",
"policy_type": "builtin",
"description": "指定虚拟私有云ID，不属于此VPC的ECS资源，视为“不合规”。",
"policy_rule_type": "dsl",
"policy_rule": {
  "allOf": [ {
    "value": "${resource().provider}",
    "comparator": "equals",
    "pattern": "ecs"
  }, {
    "value": "${resource().type}",
    "comparator": "equals",
    "pattern": "cloudservers"
  }, {
    "value": "${resource().properties.metadata.vpclid}",
    "comparator": "notEquals",
    "pattern": "${parameters('vpclid')}"
  } ]
},
"keywords": [ "ecs", "vpc" ],
"parameters": {
  "vpclid": {
    "name": null,
    "description": "VPC ID that contains the ECS instance。",
    "allowed_values": null,
    "default_value": null,
    "type": "String"
  }
}
}],
"page_info": {
  "current_count": 9,
  "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListBuiltinPolicyDefinitionsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
```

```
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    ListBuiltInPolicyDefinitionsRequest request = new ListBuiltInPolicyDefinitionsRequest();
    try {
        ListBuiltInPolicyDefinitionsResponse response = client.listBuiltInPolicyDefinitions(request);
        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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListBuiltInPolicyDefinitionsRequest()
        response = client.list_built_in_policy_definitions(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
```

```
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListBuiltInPolicyDefinitionsRequest{}  
    response, err := client.ListBuiltInPolicyDefinitions(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	参数不合法。
403	用户认证失败，或没有操作权限。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.2 查询单个内置策略

### 功能介绍

根据策略ID查询单个内置策略。



## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/policy-definitions/{policy\_definition\_id}

表 4-170 路径参数

参数	是否必选	参数类型	描述
policy_definition_id	是	String	策略ID。 最大长度： <b>36</b>

## 请求参数

表 4-171 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Language	否	String	选择接口返回的信息的语言，默认为"zh-cn"中文，可选值：zh-cn和en-us。 缺省值： <b>zh-cn</b>
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：**200**

表 4-172 响应 Body 参数

参数	参数类型	描述
id	String	策略id。
name	String	策略名称。
display_name	String	策略展示名。
policy_type	String	策略类型。
description	String	策略描述。

参数	参数类型	描述
policy_rule_type	String	策略语法类型。
policy_rule	Object	策略规则。
trigger_type	String	触发器类型，可选值："resource"和"period"。
keywords	Array of strings	关键词列表。
default_resource_types	Array of <a href="#">default_resource_types</a> objects	默认资源类型列表
parameters	Map<String, <a href="#">PolicyParameterDefinition</a> >	策略参数。

表 4-173 default\_resource\_types

参数	参数类型	描述
provider	String	云服务名称
type	String	资源类型

表 4-174 PolicyParameterDefinition

参数	参数类型	描述
name	String	策略参数名称。
description	String	策略参数描述。
allowed_values	Array of objects	策略参数允许值列表。
default_value	String	策略参数默认值。
minimum	Float	策略参数的最小值，当参数类型为Integer或Float时生效。
maximum	Float	策略参数的最大值，当参数类型为Integer或Float时生效。
min_items	Integer	策略参数的最小项数，当参数类型为Array时生效。
max_items	Integer	策略参数的最大项数，当参数类型为Array时生效。

参数	参数类型	描述
min_length	Integer	策略参数的最小字符串长度或每项的最小字符串长度，当参数类型为String或Array时生效。
max_length	Integer	策略参数的最大字符串长度或每项的最大字符串长度，当参数类型为String或Array时生效。
pattern	String	策略参数的字符串正则要求或每项的字符串正则要求，当参数类型为String或Array时生效。
type	String	策略参数类型。

状态码： 400

表 4-175 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-176 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-177 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-178 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

GET https://{endpoint}/v1/resource-manager/policy-definitions/5f8d5428ffecc14f1fb5205

## 响应示例

状态码： 200

操作成功。

```
{
  "id": "5f8d5428ffecc14f1fb5205",
  "name": "ecs-instance-no-public-ip",
  "policy_type": "builtin",
  "description": "ECS资源具有公网IP，视为“不合规”。",
  "policy_rule_type": "dsl",
  "policy_rule": {
    "allOf": [ {
      "value": "${resource().provider}",
      "comparator": "equals",
      "pattern": "ecs"
    }, {
      "value": "${resource().type}",
      "comparator": "equals",
      "pattern": "cloudservers"
    }, {
      "value": "${length(filter(resource().properties.addresses, lambda x: equals(x.OsExtIpsType, 'floating')))}",
      "comparator": "greater",
      "pattern": 0
    }
  ]
},
  "keywords": [ "ecs" ],
  "parameters": { }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowBuiltinPolicyDefinitionSolution {
```

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

    ICredential auth = new GlobalCredentials()
        .withAk(ak)
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowBuiltInPolicyDefinitionRequest request = new ShowBuiltInPolicyDefinitionRequest();
    try {
        ShowBuiltInPolicyDefinitionResponse response = client.showBuiltInPolicyDefinition(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowBuiltInPolicyDefinitionRequest()
        response = client.show_built_in_policy_definition(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或没有操作权限。
404	策略未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.3 创建合规规则

### 功能介绍

创建新的合规规则。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/resource-manager/domains/{domain\_id}/policy-assignments

表 4-179 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

### 请求参数

表 4-180 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-181 请求 Body 参数

参数	是否必选	参数类型	描述
policy_assignment_type	否	String	规则类型，包括预定义合规规则（builtin）和用户自定义合规规则（custom）
name	是	String	规则名称。 最大长度：64
description	否	String	规则描述。 最大长度：512
period	否	String	触发周期

参数	是否必选	参数类型	描述
policy_filter	否	<b>PolicyFilterDefinition</b> object	规则过滤器。
policy_definition_id	否	String	策略定义ID 最大长度： <b>36</b>
custom_policy	否	<b>CustomPolicy</b> object	自定义合规规则
parameters	否	Map<String, <b>PolicyParameter</b> >	规则参数。

表 4-182 PolicyFilterDefinition

参数	是否必选	参数类型	描述
region_id	否	String	区域ID。 最大长度： <b>128</b>
resource_provider	否	String	云服务名称。 最大长度： <b>128</b>
resource_type	否	String	资源类型。 最大长度： <b>128</b>
resource_id	否	String	资源ID。 最大长度： <b>512</b>
tag_key	否	String	标签键。 最大长度： <b>128</b>
tag_value	否	String	标签值。 最大长度： <b>256</b>

表 4-183 CustomPolicy

参数	是否必选	参数类型	描述
function_urn	是	String	自定义函数的URN 最大长度： <b>1024</b>
auth_type	是	String	自定义合规规则调用function方式
auth_value	否	Map<String, Object>	自定义合规规则调用function方式的参数值



表 4-184 PolicyParameterValue

参数	是否必选	参数类型	描述
value	否	Object	规则参数值。

## 响应参数

状态码： 200

表 4-185 响应 Body 参数

参数	参数类型	描述
policy_assignment_type	String	规则类型，包括预定义合规规则(builtin)和用户自定义合规规则(custom)
id	String	规则ID。
name	String	规则名称。
description	String	规则描述。
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
period	String	周期值，可选值：One_Hour, Three_Hours, Six_Hours, Twelve_Hours, TwentyFour_Hours
state	String	规则状态。
created	String	规则创建时间。
updated	String	规则更新时间。
policy_definition_id	String	规则的策略ID。
custom_policy	<b>CustomPolicy</b> object	自定义合规规则
parameters	Map<String, <b>PolicyParameterValue</b> >	规则参数。
created_by	String	规则的创建者

表 4-186 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度：128
resource_provider	String	云服务名称。 最大长度：128
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-187 CustomPolicy

参数	参数类型	描述
function_urn	String	自定义函数的URN 最大长度：1024
auth_type	String	自定义合规规则调用function方式
auth_value	Map<String, Object>	自定义合规规则调用function方式的参数值

表 4-188 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

状态码：400

表 4-189 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	错误消息内容。

状态码： 403

表 4-190 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-191 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

创建一个名为“allowed-images-by-id”的合规规则。

```
PUT https://{endpoint}/v1/resource-manager/domains/{domain_id}/policy-assignments
```

```
{
  "name": "allowed-images-by-id",
  "description": "The ECS resource is non-compliant if the image it used is not in the allowed list",
  "parameters": {
    "listOfAllowedImages": {
      "value": [ "ea0d6e0e-99c3-406d-a873-3bb45462b624" ]
    }
  },
  "policy_filter": {
    "resource_provider": "ecs",
    "resource_type": "cloudservers"
  },
  "policy_definition_id": "5f3c8b9a9811a13739a32ed9"
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "id": "5fb76a240f2daf0b9662a04d",
  "name": "volume-inuse-check",
}
```

```
"description": "云硬盘未挂载给任何云服务器，视为“不合规”。",
"policy_filter": {
  "region_id": null,
  "resource_provider": "evs",
  "resource_type": "volumes",
  "resource_id": null,
  "tag_key": null,
  "tag_value": null
},
"period": null,
"state": "Enabled",
"created": "2020-11-20T07:03:00",
"updated": "2020-11-20T07:03:00",
"policy_definition_id": "5fa36558aa1e6afc05a3d0a7",
"parameters": { }
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个名为“allowed-images-by-id”的合规规则。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class CreatePolicyAssignmentsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        CreatePolicyAssignmentsRequest request = new CreatePolicyAssignmentsRequest();
        PolicyAssignmentRequestBody body = new PolicyAssignmentRequestBody();
        PolicyParameterValue parametersParameters = new PolicyParameterValue();
        parametersParameters.withValue("[ea0d6e0e-99c3-406d-a873-3bb45462b624]");
        Map<String, PolicyParameterValue> listbodyParameters = new HashMap<>();
        listbodyParameters.put("listOfAllowedImages", parametersParameters);
        PolicyFilterDefinition policyFilterbody = new PolicyFilterDefinition();
        policyFilterbody.withResourceProvider("ecs")
            .withResourceType("cloudservers");
    }
}
```

```
body.withParameters(listbodyParameters);
body.withPolicyDefinitionId("5f3c8b9a9811a13739a32ed9");
body.withPolicyFilter(policyFilterbody);
body.withDescription("The ECS resource is non-compliant if the image it used is not in the allowed
list");
body.withName("allowed-images-by-id");
request.withBody(body);
try {
    CreatePolicyAssignmentsResponse response = client.createPolicyAssignments(request);
    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

创建一个名为“allowed-images-by-id”的合规规则。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CreatePolicyAssignmentsRequest()
        parametersParameters = PolicyParameterValue(
            value="[ea0d6e0e-99c3-406d-a873-3bb45462b624]"
        )
        listParametersbody = {
            "listOfAllowedImages": parametersParameters
        }
        policyFilterbody = PolicyFilterDefinition(
            resource_provider="ecs",
            resource_type="cloudservers"
        )
        request.body = PolicyAssignmentRequestBody(
            parameters=listParametersbody,
            policy_definition_id="5f3c8b9a9811a13739a32ed9",
            policy_filter=policyFilterbody,
            description="The ECS resource is non-compliant if the image it used is not in the allowed list",
            name="allowed-images-by-id"
```

```
)
response = client.create_policy_assignments(request)
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“allowed-images-by-id”的合规规则。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePolicyAssignmentsRequest{}
    var valueParameters interface{} = "[ea0d6e0e-99c3-406d-a873-3bb45462b624]"
    parametersParameters := model.PolicyParameterValue{
        Value: &valueParameters,
    }
    var listParametersbody = map[string](model.PolicyParameterValue){
        "listOfAllowedImages": parametersParameters,
    }
    resourceProviderPolicyFilter:= "ecs"
    resourceTypePolicyFilter:= "cloudservers"
    policyFilterbody := &model.PolicyFilterDefinition{
        ResourceProvider: &resourceProviderPolicyFilter,
        ResourceType: &resourceTypePolicyFilter,
    }
    policyDefinitionIdPolicyAssignmentRequestBody:= "5f3c8b9a9811a13739a32ed9"
    descriptionPolicyAssignmentRequestBody:= "The ECS resource is non-compliant if the image it used is
    not in the allowed list"
    request.Body = &model.PolicyAssignmentRequestBody{
        Parameters: listParametersbody,
        PolicyDefinitionId: &policyDefinitionIdPolicyAssignmentRequestBody,
        PolicyFilter: policyFilterbody,
        Description: &descriptionPolicyAssignmentRequestBody,
        Name: "allowed-images-by-id",
    }
    response, err := client.CreatePolicyAssignments(request)
    if err == nil {
```

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.4 列出合规规则

### 功能介绍

列出用户的合规规则。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/policy-assignments

表 4-192 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-193 Query 参数

参数	是否必选	参数类型	描述
policy_assignment_name	否	String	合规规则名称 最大长度：64
limit	否	Integer	最大的返回数量 最小值：1 最大值：500 缺省值：500
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-194 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-195 响应 Body 参数

参数	参数类型	描述
value	Array of <a href="#">PolicyAssignment</a> objects	规则列表。
page_info	<a href="#">PageInfo</a> object	分页对象。



表 4-196 PolicyAssignment

参数	参数类型	描述
policy_assignment_type	String	规则类型，包括预定义合规规则(builtin)和用户自定义合规规则(custom)
id	String	规则ID。
name	String	规则名称。
description	String	规则描述。
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
period	String	周期值，可选值：One_Hour, Three_Hours, Six_Hours, Twelve_Hours, TwentyFour_Hours
state	String	规则状态。
created	String	规则创建时间。
updated	String	规则更新时间。
policy_definition_id	String	规则的策略ID。
custom_policy	<b>CustomPolicy</b> object	自定义合规规则
parameters	Map<String, <b>PolicyParameter</b> >	规则参数。
created_by	String	规则的创建者

表 4-197 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度：128
resource_provider	String	云服务名称。 最大长度：128
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512

参数	参数类型	描述
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-198 CustomPolicy

参数	参数类型	描述
function_urn	String	自定义函数的URN 最大长度：1024
auth_type	String	自定义合规规则调用function方式
auth_value	Map<String, Object>	自定义合规规则调用function方式的参数值

表 4-199 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

表 4-200 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-201 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	错误消息内容。

状态码： 403

表 4-202 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-203 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [ {
    "id": "5fb76a240f2daf0b9662a04d",
    "name": "volume-inuse-check",
    "description": "云硬盘未挂载给任何云服务器，视为“不合规”。",
    "policy_filter": {
      "region_id": null,
      "resource_provider": "evs",
      "resource_type": "volumes",
      "resource_id": null,
      "tag_key": null,
      "tag_value": null
    },
    "state": "Enabled",
    "created": "2020-11-20T07:03:00",
    "updated": "2020-11-20T07:03:00",
    "policy_definition_id": "5fa36558aa1e6afc05a3d0a7",
    "parameters": { }
  } ],
  "page_info": {
```

```
"current_count" : 1,  
"next_marker" : 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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class ListPolicyAssignmentsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListPolicyAssignmentsRequest request = new ListPolicyAssignmentsRequest();  
        request.withPolicyAssignmentName("<policy_assignment_name>");  
        request.withLimit(<limit>);  
        request.withMarker("<marker>");  
        try {  
            ListPolicyAssignmentsResponse response = client.listPolicyAssignments(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } 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
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListPolicyAssignmentsRequest()
        request.policy_assignment_name = "<policy_assignment_name>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.list_policy_assignments(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPolicyAssignmentsRequest{
        policyAssignmentNameRequest:= "<policy_assignment_name>"
        request.PolicyAssignmentName = &policyAssignmentNameRequest
```

```
limitRequest:= int32(<limit>)  
request.Limit = &limitRequest  
markerRequest:= "<marker>"  
request.Marker = &markerRequest  
response, err := client.ListPolicyAssignments(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	认证失败或没有操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.5 更新合规规则

### 功能介绍

更新用户的合规规则。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/resource-manager/domains/{domain\_id}/policy-assignments/  
{policy\_assignment\_id}

表 4-204 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

参数	是否必选	参数类型	描述
policy_assignment_id	是	String	规则ID。 最大长度：36

## 请求参数

表 4-205 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-206 请求 Body 参数

参数	是否必选	参数类型	描述
policy_assignment_type	否	String	规则类型，包括预定义合规规则（builtin）和用户自定义合规规则（custom）
name	是	String	规则名称。 最大长度：64
description	否	String	规则描述。 最大长度：512
period	否	String	触发周期
policy_filter	否	PolicyFilterDefinition object	规则过滤器。
policy_definition_id	否	String	策略定义ID 最大长度：36
custom_policy	否	CustomPolicy object	自定义合规规则
parameters	否	Map<String, PolicyParameterValue>	规则参数。

表 4-207 PolicyFilterDefinition

参数	是否必选	参数类型	描述
region_id	否	String	区域ID。 最大长度：128
resource_provider	否	String	云服务名称。 最大长度：128
resource_type	否	String	资源类型。 最大长度：128
resource_id	否	String	资源ID。 最大长度：512
tag_key	否	String	标签键。 最大长度：128
tag_value	否	String	标签值。 最大长度：256

表 4-208 CustomPolicy

参数	是否必选	参数类型	描述
function_urn	是	String	自定义函数的URN 最大长度：1024
auth_type	是	String	自定义合规规则调用function方式
auth_value	否	Map<String,Object>	自定义合规规则调用function方式的参数值

表 4-209 PolicyParameterValue

参数	是否必选	参数类型	描述
value	否	Object	规则参数值。

## 响应参数

状态码：200



表 4-210 响应 Body 参数

参数	参数类型	描述
policy_assignment_type	String	规则类型，包括预定义合规规则(builtin)和用户自定义合规规则(custom)
id	String	规则ID。
name	String	规则名称。
description	String	规则描述。
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
period	String	周期值，可选值：One_Hour, Three_Hours, Six_Hours, Twelve_Hours, TwentyFour_Hours
state	String	规则状态。
created	String	规则创建时间。
updated	String	规则更新时间。
policy_definition_id	String	规则的策略ID。
custom_policy	<b>CustomPolicy</b> object	自定义合规规则
parameters	Map<String, <b>PolicyParameter</b> >	规则参数。
created_by	String	规则的创建者

表 4-211 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度：128
resource_provider	String	云服务名称。 最大长度：128
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512

参数	参数类型	描述
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-212 CustomPolicy

参数	参数类型	描述
function_urn	String	自定义函数的URN 最大长度：1024
auth_type	String	自定义合规规则调用function方式
auth_value	Map<String, Object>	自定义合规规则调用function方式的参数值

表 4-213 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

状态码：400

表 4-214 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-215 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-216 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-217 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
PUT https://{endpoint}/v1/resource-manager/domains/{domain_id}/policy-assignments/
{policy_assignment_id}

{
  "name" : "allowed-images-by-id",
  "description" : "The ECS resource is non-compliant if the image it used is not in the allowed list",
  "parameters" : {
    "listOfAllowedImages" : {
      "value" : [ "ea0d6e0e-99c3-406d-a873-3bb45462b624" ]
    }
  },
  "policy_filter" : {
    "resource_provider" : "ecs",
    "resource_type" : "cloudservers"
  },
  "policy_definition_id" : "5f3c8b9a9811a13739a32ed9"
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "id" : "5fb76a240f2daf0b9662a04d",
  "name" : "volume-inuse-check",
  "description" : "云硬盘未挂载给任何云服务器，视为“不合规”。",
  "policy_filter" : {
    "region_id" : null,
    "resource_provider" : "evs",
    "resource_type" : "volumes",
    "resource_id" : null,
    "tag_key" : null,
    "tag_value" : null
  },
}
```

```
"state": "Enabled",
"created": "2020-11-20T07:03:00",
"updated": "2020-11-20T07:03:00",
"policy_definition_id": "5fa36558aa1e6afc05a3d0a7",
"parameters": { }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdatePolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdatePolicyAssignmentRequest request = new UpdatePolicyAssignmentRequest();
        PolicyAssignmentRequestBody body = new PolicyAssignmentRequestBody();
        PolicyParameterValue parametersParameters = new PolicyParameterValue();
        parametersParameters.withValue("[ea0d6e0e-99c3-406d-a873-3bb45462b624]");
        Map<String, PolicyParameterValue> listbodyParameters = new HashMap<>();
        listbodyParameters.put("listOfAllowedImages", parametersParameters);
        PolicyFilterDefinition policyFilterbody = new PolicyFilterDefinition();
        policyFilterbody.withResourceProvider("ecs")
            .withResourceType("cloudservers");
        body.withParameters(listbodyParameters);
        body.withPolicyDefinitionId("5f3c8b9a9811a13739a32ed9");
        body.withPolicyFilter(policyFilterbody);
        body.withDescription("The ECS resource is non-compliant if the image it used is not in the allowed
list");
        body.withName("allowed-images-by-id");
        request.withBody(body);
        try {
            UpdatePolicyAssignmentResponse response = client.updatePolicyAssignment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = UpdatePolicyAssignmentRequest()
        parametersParameters = PolicyParameterValue(
            value="[ea0d6e0e-99c3-406d-a873-3bb45462b624]"
        )
        listParametersbody = {
            "listOfAllowedImages": parametersParameters
        }
        policyFilterbody = PolicyFilterDefinition(
            resource_provider="ecs",
            resource_type="cloudservers"
        )
        request.body = PolicyAssignmentRequestBody(
            parameters=listParametersbody,
            policy_definition_id="5f3c8b9a9811a13739a32ed9",
            policy_filter=policyFilterbody,
            description="The ECS resource is non-compliant if the image it used is not in the allowed list",
            name="allowed-images-by-id"
        )
        response = client.update_policy_assignment(request)
        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/global"
config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePolicyAssignmentRequest{
        var valueParameters interface{} = "[ea0d6e0e-99c3-406d-a873-3bb45462b624]"
        parametersParameters := model.PolicyParameterValue{
            Value: &valueParameters,
        }
        var listParametersbody = map[string](model.PolicyParameterValue){
            "listOfAllowedImages": parametersParameters,
        }
        resourceProviderPolicyFilter:= "ecs"
        resourceTypePolicyFilter:= "cloudservers"
        policyFilterbody := &model.PolicyFilterDefinition{
            ResourceProvider: &resourceProviderPolicyFilter,
            ResourceType: &resourceTypePolicyFilter,
        }
        policyDefinitionIdPolicyAssignmentRequestBody:= "5f3c8b9a9811a13739a32ed9"
        descriptionPolicyAssignmentRequestBody:= "The ECS resource is non-compliant if the image it used is
        not in the allowed list"
        request.Body = &model.PolicyAssignmentRequestBody{
            Parameters: listParametersbody,
            PolicyDefinitionId: &policyDefinitionIdPolicyAssignmentRequestBody,
            PolicyFilter: policyFilterbody,
            Description: &descriptionPolicyAssignmentRequestBody,
            Name: "allowed-images-by-id",
        }
    }
    response, err := client.UpdatePolicyAssignment(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或没有操作权限。
404	合规规则未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.6 获取单个合规规则

### 功能介绍

根据规则ID获取单个规则。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/policy-assignments/  
{policy\_assignment\_id}

表 4-218 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
policy_assignment_id	是	String	规则ID。 最大长度：36

## 请求参数

表 4-219 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 200

表 4-220 响应 Body 参数

参数	参数类型	描述
policy_assignment_type	String	规则类型，包括预定义合规规则(builtin)和用户自定义合规规则(custom)
id	String	规则ID。
name	String	规则名称。
description	String	规则描述。
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
period	String	周期值，可选值：One_Hour, Three_Hours, Six_Hours, Twelve_Hours, TwentyFour_Hours
state	String	规则状态。
created	String	规则创建时间。
updated	String	规则更新时间。
policy_definition_id	String	规则的策略ID。
custom_policy	<b>CustomPolicy</b> object	自定义合规规则
parameters	Map<String, <b>PolicyParameter</b> >	规则参数。
created_by	String	规则的创建者



表 4-221 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度：128
resource_provider	String	云服务名称。 最大长度：128
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-222 CustomPolicy

参数	参数类型	描述
function_urn	String	自定义函数的URN 最大长度：1024
auth_type	String	自定义合规规则调用function方式
auth_value	Map<String, Object>	自定义合规规则调用function方式的参数值

表 4-223 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

状态码：400

表 4-224 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	错误消息内容。

状态码： 403

表 4-225 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-226 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-227 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{  
  "id": "5fb76a240f2daf0b9662a04d",  
  "name": "volume-inuse-check",  
  "description": "云硬盘未挂载给任何云服务器，视为“不合规”。",  
}
```

```
"policy_filter" : {
  "region_id" : null,
  "resource_provider" : "evs",
  "resource_type" : "volumes",
  "resource_id" : null,
  "tag_key" : null,
  "tag_value" : null
},
"state" : "Enabled",
"created" : "2020-11-20T07:03:00",
"updated" : "2020-11-20T07:03:00",
"policy_definition_id" : "5fa36558aa1e6afc05a3d0a7",
"parameters" : { }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowPolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowPolicyAssignmentRequest request = new ShowPolicyAssignmentRequest();
        try {
            ShowPolicyAssignmentResponse response = client.showPolicyAssignment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkconfig.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = GlobalCredentials(ak, sk) \  
  
    client = ConfigClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ShowPolicyAssignmentRequest()  
        response = client.show_policy_assignment(request)  
        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/global"  
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())
```

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或没有操作权限。
404	合规规则未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.7 删除合规规则

### 功能介绍

根据规则ID删除此规则。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/resource-manager/domains/{domain\_id}/policy-assignments/  
{policy\_assignment\_id}

表 4-228 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
policy_assignment_id	是	String	规则ID。 最大长度：36

## 请求参数

表 4-229 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：400

表 4-230 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-231 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-232 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-233 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeletePolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
```

```
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    DeletePolicyAssignmentRequest request = new DeletePolicyAssignmentRequest();
    try {
        DeletePolicyAssignmentResponse response = client.deletePolicyAssignment(request);
        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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeletePolicyAssignmentRequest()
        response = client.delete_policy_assignment(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
```



```
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.DeletePolicyAssignmentRequest{}  
    response, err := client.DeletePolicyAssignment(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或没有操作权限。
404	合规规则未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.8 启用合规规则

### 功能介绍

根据规则ID启用此规则。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v1/resource-manager/domains/{domain\_id}/policy-assignments/  
{policy\_assignment\_id}/enable

表 4-234 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
policy_assignment_id	是	String	规则ID。 最大长度：36

## 请求参数

表 4-235 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：400

表 4-236 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-237 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-238 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-239 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;
```

```
public class EnablePolicyAssignmentSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        EnablePolicyAssignmentRequest request = new EnablePolicyAssignmentRequest();  
        try {  
            EnablePolicyAssignmentResponse response = client.enablePolicyAssignment(request);  
            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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkconfig.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = GlobalCredentials(ak, sk) \  
  
    client = ConfigClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = EnablePolicyAssignmentRequest()  
        response = client.enable_policy_assignment(request)  
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
202	操作接受。
400	输入参数不合法。
403	用户认证失败或没有操作权限。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.9 停用合规规则

### 功能介绍

根据规则ID停用此规则。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/domains/{domain\_id}/policy-assignments/  
{policy\_assignment\_id}/disable

表 4-240 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
policy_assignment_id	是	String	规则ID。 最大长度：36

### 请求参数

表 4-241 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

### 响应参数

状态码： 400

表 4-242 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-243 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-244 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-245 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DisablePolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        DisablePolicyAssignmentRequest request = new DisablePolicyAssignmentRequest();
        try {
            DisablePolicyAssignmentResponse response = client.disablePolicyAssignment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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



```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = DisablePolicyAssignmentRequest()
    response = client.disable_policy_assignment(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
202	操作接受。
400	输入参数不合法。
403	用户认证失败或没有操作权限。
404	合规规则未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.5.10 运行合规评估

#### 功能介绍

根据规则ID评估此规则。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/resource-manager/domains/{domain\_id}/policy-assignments/{policy\_assignment\_id}/policy-states/run-evaluation

表 4-246 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
policy_assignment_id	是	String	规则ID。 最大长度：36

## 请求参数

表 4-247 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 400

表 4-248 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-249 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-250 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-251 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class RunEvaluationByPolicyAssignmentIdSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        RunEvaluationByPolicyAssignmentIdRequest request = new
        RunEvaluationByPolicyAssignmentIdRequest();
        try {
            RunEvaluationByPolicyAssignmentIdResponse response =
            client.runEvaluationByPolicyAssignmentId(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
e.printStackTrace();
} 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = RunEvaluationByPolicyAssignmentIdRequest()
        response = client.run_evaluation_by_policy_assignment_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
```

```
WithSk(sk).
Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
202	操作接受。
400	输入参数不合法。
403	认证失败或没有操作权限。
404	合规规则未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.5.11 获取规则的评估状态

#### 功能介绍

根据规则ID查询此规则的评估状态。

#### 调用方法

请参见[如何调用API](#)。

#### URI

```
GET /v1/resource-manager/domains/{domain_id}/policy-assignments/
{policy_assignment_id}/policy-states/evaluation-state
```

表 4-252 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
policy_assignment_id	是	String	规则ID。 最大长度：36

## 请求参数

表 4-253 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-254 响应 Body 参数

参数	参数类型	描述
policy_assignment_id	String	规则ID。
state	String	评估任务执行状态。
start_time	String	评估任务开始时间。
end_time	String	评估任务结束时间。
error_message	String	评估任务失败信息。

状态码：400

表 4-255 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-256 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-257 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-258 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{  
  "state": "Succeeded",
```



```
"policy_assignment_id" : "5fb6292765ee7f5e92a7ca4b",  
"start_time" : "2020-11-19T08:13:27.441Z",  
"end_time" : "2020-11-19T08:13:27.485Z",  
"error_message" : 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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class ShowEvaluationStateByAssignmentIdSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowEvaluationStateByAssignmentIdRequest request = new  
        ShowEvaluationStateByAssignmentIdRequest();  
        try {  
            ShowEvaluationStateByAssignmentIdResponse response =  
            client.showEvaluationStateByAssignmentId(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } 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
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowEvaluationStateByAssignmentIdRequest()
        response = client.show_evaluation_state_by_assignment_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

```
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	认证失败或没有权限。
404	合规规则未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.12 获取资源的合规结果

### 功能介绍

根据资源ID查询所有合规结果。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/resources/{resource\_id}/policy-states

表 4-259 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
resource_id	是	String	资源ID。 最大长度：512

表 4-260 Query 参数

参数	是否必选	参数类型	描述
compliance_state	否	String	合规结果。 最大长度：16
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-261 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-262 响应 Body 参数

参数	参数类型	描述
value	Array of <b>PolicyState</b> objects	合规结果查询返回值。
page_info	<b>PageInfo</b> object	分页对象。

表 4-263 PolicyState

参数	参数类型	描述
domain_id	String	用户ID。
region_id	String	资源区域ID。
resource_id	String	资源ID。
resource_name	String	资源名称。
resource_provider	String	云服务名称。
resource_type	String	资源类型。
trigger_type	String	触发器类型，可选值为"resource"和"period"。
compliance_state	String	合规结果。
policy_assignment_id	String	规则ID。
policy_assignment_name	String	规则名称。
policy_definition_id	String	策略ID。
evaluation_time	String	合规状态评估时间。

表 4-264 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码：**400**

表 4-265 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-266 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-267 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-268 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{  
  "value": [ {
```

```
"domain_id" : "059b5c937100d3e40ff0c00a7675a0a0",
"region_id" : "regionid1",
"resource_id" : "010d95bd-87cd-4f22-ac00-db7fba7d927e",
"resource_name" : "ecs-cc-image-test1",
"resource_provider" : "ecs",
"resource_type" : "cloudservers",
"compliance_state" : "NonCompliant",
"policy_assignment_id" : "5fb618a726a24c53767fa049",
"policy_assignment_name" : "policy-assignment-test1",
"policy_definition_id" : "5fa265c0aa1e6afc05a0ff07",
"evaluation_time" : 1605776482523
}],
"page_info" : {
  "current_count" : 1,
  "next_marker" : 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListPolicyStatesByResourceIdSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListPolicyStatesByResourceIdRequest request = new ListPolicyStatesByResourceIdRequest();
        request.withComplianceState("<compliance_state>");
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        try {
            ListPolicyStatesByResourceIdResponse response = client.listPolicyStatesByResourceId(request);
            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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListPolicyStatesByResourceIdRequest()
        request.compliance_state = "<compliance_state>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.list_policy_states_by_resource_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
```



```
WithSk(sk).
Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListPolicyStatesByResourceIdRequest{}
complianceStateRequest:= "<compliance_state>"
request.ComplianceState = &complianceStateRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
response, err := client.ListPolicyStatesByResourceId(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	认证失败或没有操作权限。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.5.13 获取规则的合规结果

#### 功能介绍

根据规则ID查询所有的合规结果。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/policy-assignments/  
{policy\_assignment\_id}/policy-states

表 4-269 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
policy_assignment_id	是	String	规则ID。 最大长度：36

表 4-270 Query 参数

参数	是否必选	参数类型	描述
compliance_state	否	String	合规结果。 最大长度：16
resource_id	否	String	资源ID。 最大长度：512
resource_name	否	String	资源名称。 最大长度：256
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-271 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 200

表 4-272 响应 Body 参数

参数	参数类型	描述
value	Array of <b>PolicyState</b> objects	合规结果查询返回值。
page_info	<b>PageInfo</b> object	分页对象。

表 4-273 PolicyState

参数	参数类型	描述
domain_id	String	用户ID。
region_id	String	资源区域ID。
resource_id	String	资源ID。
resource_name	String	资源名称。
resource_provider	String	云服务名称。
resource_type	String	资源类型。
trigger_type	String	触发器类型，可选值为"resource"和"period"。
compliance_state	String	合规结果。
policy_assignment_id	String	规则ID。
policy_assignment_name	String	规则名称。

参数	参数类型	描述
policy_definition_id	String	策略ID。
evaluation_time	String	合规状态评估时间。

表 4-274 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-275 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-276 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-277 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-278 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [ {
    "domain_id": "059b5c937100d3e40ff0c00a7675a0a0",
    "region_id": "regionid1",
    "resource_id": "010d95bd-87cd-4f22-ac00-db7fba7d927e",
    "resource_name": "ecs-cc-image-test1",
    "resource_provider": "ecs",
    "resource_type": "cloudservers",
    "compliance_state": "NonCompliant",
    "policy_assignment_id": "5fb618a726a24c53767fa049",
    "policy_assignment_name": "policy-assignment-test1",
    "policy_definition_id": "5fa265c0aa1e6afc05a0ff07",
    "evaluation_time": 1605776482523
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListPolicyStatesByAssignmentIdSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListPolicyStatesByAssignmentIdRequest request = new ListPolicyStatesByAssignmentIdRequest();
        request.withComplianceState("<compliance_state>");
        request.withResourceId("<resource_id>");
        request.withResourceName("<resource_name>");
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        try {
            ListPolicyStatesByAssignmentIdResponse response = client.listPolicyStatesByAssignmentId(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

```
credentials = GlobalCredentials(ak, sk) \

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

try:
    request = ListPolicyStatesByAssignmentIdRequest()
    request.compliance_state = "<compliance_state>"
    request.resource_id = "<resource_id>"
    request.resource_name = "<resource_name>"
    request.limit = <limit>
    request.marker = "<marker>"
    response = client.list_policy_states_by_assignment_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPolicyStatesByAssignmentIdRequest{}
    complianceStateRequest := "<compliance_state>"
    request.ComplianceState = &complianceStateRequest
    resourceIdRequest := "<resource_id>"
    request.ResourceId = &resourceIdRequest
    resourceNameRequest := "<resource_name>"
    request.ResourceName = &resourceNameRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListPolicyStatesByAssignmentId(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```

```
    fmt.Println(err)
  }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或没有操作权限。
404	合规规则未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.14 获取用户的合规结果

### 功能介绍

查询用户所有的合规结果。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/policy-states

表 4-279 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36



表 4-280 Query 参数

参数	是否必选	参数类型	描述
compliance_state	否	String	合规结果。 最大长度：16
resource_id	否	String	资源ID。 最大长度：512
resource_name	否	String	资源名称。 最大长度：256
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-281 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-282 响应 Body 参数

参数	参数类型	描述
value	Array of <b>PolicyState</b> objects	合规结果查询返回值。

参数	参数类型	描述
page_info	<b>PageInfo</b> object	分页对象。

表 4-283 PolicyState

参数	参数类型	描述
domain_id	String	用户ID。
region_id	String	资源区域ID。
resource_id	String	资源ID。
resource_name	String	资源名称。
resource_provider	String	云服务名称。
resource_type	String	资源类型。
trigger_type	String	触发器类型，可选值为"resource"和"period"。
compliance_state	String	合规结果。
policy_assignment_id	String	规则ID。
policy_assignment_name	String	规则名称。
policy_definition_id	String	策略ID。
evaluation_time	String	合规状态评估时间。

表 4-284 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-285 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-286 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-287 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-288 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [ {
    "domain_id": "059b5c937100d3e40ff0c00a7675a0a0",
    "region_id": "regionid1",
    "resource_id": "010d95bd-87cd-4f22-ac00-db7fba7d927e",
    "resource_name": "ecs-cc-image-test1",
    "resource_provider": "ecs",
    "resource_type": "cloudservers",
    "compliance_state": "NonCompliant",
    "policy_assignment_id": "5fb618a726a24c53767fa049",
    "policy_assignment_name": "policy-assignment-test1",
    "policy_definition_id": "5fa265c0aa1e6afc05a0ff07",
    "evaluation_time": 1605776482523
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListPolicyStatesByDomainIdSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListPolicyStatesByDomainIdRequest request = new ListPolicyStatesByDomainIdRequest();
        request.withComplianceState("<compliance_state>");
        request.withResourceId("<resource_id>");
        request.withResourceName("<resource_name>");
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        try {
            ListPolicyStatesByDomainIdResponse response = client.listPolicyStatesByDomainId(request);
            System.out.println(response.toString());
        }
    }
}
```

```
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListPolicyStatesByDomainIdRequest()
        request.compliance_state = "<compliance_state>"
        request.resource_id = "<resource_id>"
        request.resource_name = "<resource_name>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.list_policy_states_by_domain_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

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

auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListPolicyStatesByDomainIdRequest{
    complianceStateRequest:= "<compliance_state>"
    request.ComplianceState = &complianceStateRequest
    resourceIdRequest:= "<resource_id>"
    request.ResourceId = &resourceIdRequest
    resourceNameRequest:= "<resource_name>"
    request.ResourceName = &resourceNameRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListPolicyStatesByDomainId(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或没有操作权限。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.15 更新合规评估结果

### 功能介绍

更新用户自定义合规规则的合规评估结果。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/resource-manager/domains/{domain\_id}/policy-states

表 4-289 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

### 请求参数

表 4-290 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-291 请求 Body 参数

参数	是否必选	参数类型	描述
policy_resource	是	<b>PolicyResource</b> object	资源。
trigger_type	是	String	触发器类型。
compliance_state	是	String	合规结果。
policy_assignment_id	是	String	规则ID。

参数	是否必选	参数类型	描述
policy_assignment_name	否	String	规则名称。
evaluation_time	是	String	合规状态评估时间。
evaluation_hash	是	String	评估校验码。

表 4-292 PolicyResource

参数	是否必选	参数类型	描述
resource_id	否	String	资源ID。
resource_name	否	String	资源名称。
resource_provider	否	String	云服务名称。
resource_type	否	String	资源类型。
region_id	否	String	区域ID。
domain_id	否	String	资源所属用户ID。

## 响应参数

状态码： 200

表 4-293 响应 Body 参数

参数	参数类型	描述
domain_id	String	用户ID。
region_id	String	资源区域ID。
resource_id	String	资源ID。
resource_name	String	资源名称。
resource_provider	String	云服务名称。
resource_type	String	资源类型。
trigger_type	String	触发器类型，可选值为"resource"和"period"。



参数	参数类型	描述
compliance_state	String	合规结果。
policy_assignment_id	String	规则ID。
policy_assignment_name	String	规则名称。
policy_definition_id	String	策略ID。
evaluation_time	String	合规状态评估时间。

状态码： 400

表 4-294 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-295 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-296 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

functionGraph上报合规评估结果为“NonCompliant”。

```
PUT https://{endpoint}/v1/resource-manager/domains/{domain_id}/policy-states
```

```
{
  "policy_resource": {
    "domain_id": "d0123456789",
    "region_id": "global",
    "resource_id": "abc0123456789",
    "resource_name": "test_user",
    "resource_provider": "iam",
    "resource_type": "users"
  },
  "trigger_type": "resource",
  "compliance_state": "NonCompliant",
  "policy_assignment_id": "abc0123456789abc",
  "policy_assignment_name": "custom_policy",
  "evaluation_time": 1667374060248,
  "evaluation_hash": "89342b8f338165651991afb8bd471396"
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "domain_id": "d0123456789",
  "region_id": "global",
  "resource_id": "abc0123456789",
  "resource_name": "test_user",
  "resource_provider": "iam",
  "resource_type": "users",
  "trigger_type": "resource",
  "compliance_state": "NonCompliant",
  "policy_assignment_id": "abc0123456789abc",
  "policy_assignment_name": "custom_policy",
  "policy_definition_id": null,
  "evaluation_time": 1667374060248
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

functionGraph上报合规评估结果为“NonCompliant”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class UpdatePolicyStateSolution {
```

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

    ICredential auth = new GlobalCredentials()
        .withAk(ak)
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    UpdatePolicyStateRequest request = new UpdatePolicyStateRequest();
    PolicyStateRequestBody body = new PolicyStateRequestBody();
    PolicyResource policyResourcebody = new PolicyResource();
    policyResourcebody.withResourceId("abc0123456789")
        .withResourceName("test_user")
        .withResourceProvider("iam")
        .withResourceType("users")
        .withRegionId("global")
        .withDomainId("d0123456789");
    body.withEvaluationHash("89342b8f338165651991afb8bd471396");
    body.withEvaluationTime("1667374060248");
    body.withPolicyAssignmentName("custom_policy");
    body.withPolicyAssignmentId("abc0123456789abc");

    body.withComplianceState(PolicyStateRequestBody.ComplianceStateEnum.fromValue("NonCompliant"));
    body.withTriggerType(PolicyStateRequestBody.TriggerTypeEnum.fromValue("resource"));
    body.withPolicyResource(policyResourcebody);
    request.withBody(body);
    try {
        UpdatePolicyStateResponse response = client.updatePolicyState(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

functionGraph上报告合规评估结果为“NonCompliant”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = UpdatePolicyStateRequest()
    policyResourcebody = PolicyResource(
        resource_id="abc0123456789",
        resource_name="test_user",
        resource_provider="iam",
        resource_type="users",
        region_id="global",
        domain_id="d0123456789"
    )
    request.body = PolicyStateRequestBody(
        evaluation_hash="89342b8f338165651991afb8bd471396",
        evaluation_time="1667374060248",
        policy_assignment_name="custom_policy",
        policy_assignment_id="abc0123456789abc",
        compliance_state="NonCompliant",
        trigger_type="resource",
        policy_resource=policyResourcebody
    )
    response = client.update_policy_state(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

functionGraph上报告合规评估结果为“NonCompliant”。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.UpdatePolicyStateRequest{  
    resourceIdPolicyResource:= "abc0123456789"  
    resourceNamePolicyResource:= "test_user"  
    resourceProviderPolicyResource:= "iam"  
    resourceTypePolicyResource:= "users"  
    regionIdPolicyResource:= "global"  
    domainIdPolicyResource:= "d0123456789"  
    policyResourcebody := &model.PolicyResource{  
        ResourceId: &resourceIdPolicyResource,  
        ResourceName: &resourceNamePolicyResource,  
        ResourceProvider: &resourceProviderPolicyResource,  
        ResourceType: &resourceTypePolicyResource,  
        RegionId: &regionIdPolicyResource,  
        DomainId: &domainIdPolicyResource,  
    }  
    policyAssignmentNamePolicyStateRequestBody:= "custom_policy"  
    request.Body = &model.PolicyStateRequestBody{  
        EvaluationHash: "89342b8f338165651991afb8bd471396",  
        EvaluationTime: "1667374060248",  
        PolicyAssignmentName: &policyAssignmentNamePolicyStateRequestBody,  
        PolicyAssignmentId: "abc0123456789abc",  
        ComplianceState: model.GetPolicyStateRequestBodyComplianceStateEnum().NON_COMPLIANT,  
        TriggerType: model.GetPolicyStateRequestBodyTriggerTypeEnum().RESOURCE,  
        PolicyResource: policyResourcebody,  
    }  
    response, err := client.UpdatePolicyState(request)  
    if err == nil {  
        fmt.Printf("%v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.16 创建组织合规规则

### 功能介绍

创建组织合规规则，如果规则名称已存在，则为更新操作。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/resource-manager/organizations/{organization\_id}/policy-assignments

表 4-297 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

### 请求参数

表 4-298 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-299 请求 Body 参数

参数	是否必选	参数类型	描述
excluded_accounts	否	Array of strings	需要排除配置规则的账号。 最大长度：32
organization_policy_assignment_name	是	String	组织合规规则名称。 最大长度：60
managed_policy_assignment_metadata	否	<b>ManagedPolicyAssignmentMetadata</b> object	托管规则元数据。

表 4-300 ManagedPolicyAssignmentMetadata

参数	是否必选	参数类型	描述
description	否	String	规则描述。 最大长度：512
period	否	String	触发周期。
parameters	否	Map<String,PolicyParameterValue>	输入参数。
policy_filter	否	PolicyFilterDefinition object	规则过滤器。
policy_definition_id	是	String	预定义策略标识符。 最大长度：36

表 4-301 PolicyParameterValue

参数	是否必选	参数类型	描述
value	否	Object	规则参数值。

表 4-302 PolicyFilterDefinition

参数	是否必选	参数类型	描述
region_id	否	String	区域ID。 最大长度：128
resource_provider	否	String	云服务名称。 最大长度：128
resource_type	否	String	资源类型。 最大长度：128
resource_id	否	String	资源ID。 最大长度：512
tag_key	否	String	标签键。 最大长度：128
tag_value	否	String	标签值。 最大长度：256

## 响应参数

状态码： 200

表 4-303 响应 Body 参数

参数	参数类型	描述
owner_id	String	组织合规规则创建者。
organization_id	String	组织ID。
organization_policy_assignment_urn	String	组织合规规则资源唯一标识。
organization_policy_assignment_id	String	组织合规规则ID。
organization_policy_assignment_name	String	组织合规规则名称。
description	String	描述信息。
period	String	触发周期。
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
parameters	Map<String, <b>PolicyParameter</b> >	规则参数。
policy_definition_id	String	策略ID。
created_at	String	创建时间。
updated_at	String	更新时间。

表 4-304 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度： 128
resource_provider	String	云服务名称。 最大长度： 128



参数	参数类型	描述
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-305 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

状态码：400

表 4-306 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-307 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-308 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

创建一个名为“allowed-images-by-id”的组织合规规则。

PUT https://{endpoint}/v1/resource-manager/organizations/{organization\_id}/policy-assignments

```
{
  "organization_policy_assignment_name": "allowed-images-by-id",
  "managed_policy_assignment_metadata": {
    "description": "The ECS resource is non-compliant if the image it used is not in the allowed list.",
    "parameters": {
      "listOfAllowedImages": {
        "value": [ "ea0d6e0e-99c3-406d-a873-3bb45462b624" ]
      }
    },
    "policy_filter": {
      "resource_provider": "ecs",
      "resource_type": "cloudservers"
    },
    "policy_definition_id": "5fa265c0aa1e6afc05a0ff07"
  }
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "owner_id": "e74e043fab784a45ad88f5ef6a4bcffc",
  "organization_id": "o-doxbpkzqsdd51ti1k27w2y8fitakrkn",
  "organization_policy_assignment_urn":
  "rms::e74e043fab784a45ad88f5ef6a4bcffc:organizationPolicyAssignment:o-
  doxbpkzqsdd51ti1k27w2y8fitakrkn/d02b7fa9d5a74e638c1402d0868f71fd",
  "organization_policy_assignment_id": "d02b7fa9d5a74e638c1402d0868f71fd",
  "organization_policy_assignment_name": "allowed-images-by-id",
  "description": "The ECS resource is non-compliant if the image it used is not in the allowed list.",
  "period": null,
  "policy_filter": {
    "region_id": null,
    "resource_provider": "evs",
    "resource_type": "volumes",
    "resource_id": null,
    "tag_key": null,
    "tag_value": null
  },
  "policy_definition_id": "5fa265c0aa1e6afc05a0ff07",
  "parameters": { },
  "created_at": "2022-01-18T06:37:54.432Z",
  "updated_at": "2022-01-18T06:37:54.432Z"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个名为“allowed-images-by-id”的组织合规规则。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class CreateOrganizationPolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

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

        CreateOrganizationPolicyAssignmentRequest request = new
        CreateOrganizationPolicyAssignmentRequest();
        OrganizationPolicyAssignmentRequest body = new OrganizationPolicyAssignmentRequest();
        PolicyFilterDefinition policyFilterManagedPolicyAssignmentMetadata = new PolicyFilterDefinition();
        policyFilterManagedPolicyAssignmentMetadata.withResourceProvider("ecs")
            .withResourceType("cloudservers");
        PolicyParameterValue parametersParameters = new PolicyParameterValue();
        parametersParameters.withValue("[ea0d6e0e-99c3-406d-a873-3bb45462b624]");
        Map<String, PolicyParameterValue> listManagedPolicyAssignmentMetadataParameters = new
        HashMap<>();
        listManagedPolicyAssignmentMetadataParameters.put("listOfAllowedImages", parametersParameters);
        ManagedPolicyAssignmentMetadata managedPolicyAssignmentMetadatabody = new
        ManagedPolicyAssignmentMetadata();
        managedPolicyAssignmentMetadatabody.withDescription("The ECS resource is non-compliant if the
        image it used is not in the allowed list.")
            .withParameters(listManagedPolicyAssignmentMetadataParameters)
            .withPolicyFilter(policyFilterManagedPolicyAssignmentMetadata)
            .withPolicyDefinitionId("5fa265c0aa1e6afc05a0ff07");
        body.withManagedPolicyAssignmentMetadata(managedPolicyAssignmentMetadatabody);
        body.withOrganizationPolicyAssignmentName("allowed-images-by-id");
        request.withBody(body);
        try {
            CreateOrganizationPolicyAssignmentResponse response =
            client.createOrganizationPolicyAssignment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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“allowed-images-by-id”的组织合规规则。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CreateOrganizationPolicyAssignmentRequest()
        policyFilterManagedPolicyAssignmentMetadata = PolicyFilterDefinition(
            resource_provider="ecs",
            resource_type="cloudservers"
        )
        parametersParameters = PolicyParameterValue(
            value="[ea0d6e0e-99c3-406d-a873-3bb45462b624]"
        )
        listParametersManagedPolicyAssignmentMetadata = {
            "listOfAllowedImages": parametersParameters
        }
        managedPolicyAssignmentMetadatabody = ManagedPolicyAssignmentMetadata(
            description="The ECS resource is non-compliant if the image it used is not in the allowed list.",
            parameters=listParametersManagedPolicyAssignmentMetadata,
            policy_filter=policyFilterManagedPolicyAssignmentMetadata,
            policy_definition_id="5fa265c0aa1e6afc05a0ff07"
        )
        request.body = OrganizationPolicyAssignmentRequest(
            managed_policy_assignment_metadata=managedPolicyAssignmentMetadatabody,
            organization_policy_assignment_name="allowed-images-by-id"
        )
        response = client.create_organization_policy_assignment(request)
        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“allowed-images-by-id”的组织合规规则。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateOrganizationPolicyAssignmentRequest{
        resourceProviderPolicyFilter:= "ecs"
        resourceTypePolicyFilter:= "cloudservers"
        policyFilterManagedPolicyAssignmentMetadata := &model.PolicyFilterDefinition{
            ResourceProvider: &resourceProviderPolicyFilter,
            ResourceType: &resourceTypePolicyFilter,
        }
        var valueParameters interface{} = "[ea0d6e0e-99c3-406d-a873-3bb45462b624]"
        parametersParameters := model.PolicyParameterValue{
            Value: &valueParameters,
        }
        var listParametersManagedPolicyAssignmentMetadata = map[string](model.PolicyParameterValue){
            "listOfAllowedImages": parametersParameters,
        }
        descriptionManagedPolicyAssignmentMetadata:= "The ECS resource is non-compliant if the image it
        used is not in the allowed list."
        managedPolicyAssignmentMetadatabody := &model.ManagedPolicyAssignmentMetadata{
            Description: &descriptionManagedPolicyAssignmentMetadata,
            Parameters: listParametersManagedPolicyAssignmentMetadata,
            PolicyFilter: policyFilterManagedPolicyAssignmentMetadata,
            PolicyDefinitionId: "5fa265c0aa1e6afc05a0ff07",
        }
        request.Body = &model.OrganizationPolicyAssignmentRequest{
            ManagedPolicyAssignmentMetadata: managedPolicyAssignmentMetadatabody,
            OrganizationPolicyAssignmentName: "allowed-images-by-id",
        }
        response, err := client.CreateOrganizationPolicyAssignment(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.17 查询组织合规规则列表

### 功能介绍

查询组织合规规则列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/organizations/{organization\_id}/policy-assignments

表 4-309 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

表 4-310 Query 参数

参数	是否必选	参数类型	描述
organization_policy_assignment_id	否	String	组织合规规则ID 最大长度：512

参数	是否必选	参数类型	描述
organization_policy_assignment_name	否	String	组织合规规则名称。 最大长度：60
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-311 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-312 响应 Body 参数

参数	参数类型	描述
organization_policy_assignments	Array of <b>OrganizationPolicyAssignmentResponse</b> objects	组织合规规则列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-313 OrganizationPolicyAssignmentResponse

参数	参数类型	描述
owner_id	String	组织合规规则创建者。
organization_id	String	组织ID。
organization_policy_assignment_urn	String	组织合规规则资源唯一标识。
organization_policy_assignment_id	String	组织合规规则ID。
organization_policy_assignment_name	String	组织合规规则名称。
description	String	描述信息。
period	String	触发周期。
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
parameters	Map<String, <b>PolicyParameter</b> Value>	规则参数。
policy_definition_id	String	策略ID。
created_at	String	创建时间。
updated_at	String	更新时间。

表 4-314 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度：128
resource_provider	String	云服务名称。 最大长度：128
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512



参数	参数类型	描述
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-315 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

表 4-316 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-317 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-318 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-319 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "organization_policy_assignments": [ {
    "owner_id": "e74e043fab784a45ad88f5ef6a4bcffc",
    "organization_id": "o-doxbpkzqsdd51ti1k27w2y8fitakrknp",
    "organization_policy_assignment_urn":
    "rms::e74e043fab784a45ad88f5ef6a4bcffc:organizationPolicyAssignment:o-
    doxbpkzqsdd51ti1k27w2y8fitakrknp/d02b7fa9d5a74e638c1402d0868f71fd",
    "organization_policy_assignment_id": "d02b7fa9d5a74e638c1402d0868f71fd",
    "organization_policy_assignment_name": "allowed-images-by-id",
    "description": "The ECS resource is non-compliant if the image it used is not in the allowed list",
    "period": null,
    "policy_filter": {
      "region_id": null,
      "resource_provider": "evs",
      "resource_type": "volumes",
      "resource_id": null,
      "tag_key": null,
      "tag_value": null
    },
    "policy_definition_id": "5fa265c0aa1e6afc05a0ff07",
    "parameters": { },
    "created_at": "2022-01-18T06:37:54.432Z",
    "updated_at": "2022-01-18T06:37:54.432Z"
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": "CAESIGQwMmi3ZmE5ZDVhNzRlNjM4YzE0MDJkMDg2OGY3MWZkGgTnpS5u"
  }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListOrganizationPolicyAssignmentsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListOrganizationPolicyAssignmentsRequest request = new ListOrganizationPolicyAssignmentsRequest();
        request.withOrganizationPolicyAssignmentId("<organization_policy_assignment_id>");
        request.withOrganizationPolicyAssignmentName("<organization_policy_assignment_name>");
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        try {
            ListOrganizationPolicyAssignmentsResponse response =
client.listOrganizationPolicyAssignments(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

```
credentials = GlobalCredentials(ak, sk) \

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

try:
    request = ListOrganizationPolicyAssignmentsRequest()
    request.organization_policy_assignment_id = "<organization_policy_assignment_id>"
    request.organization_policy_assignment_name = "<organization_policy_assignment_name>"
    request.limit = <limit>
    request.marker = "<marker>"
    response = client.list_organization_policy_assignments(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListOrganizationPolicyAssignmentsRequest{
        organizationPolicyAssignmentIdRequest:= "<organization_policy_assignment_id>"
        request.OrganizationPolicyAssignmentId = &organizationPolicyAssignmentIdRequest
        organizationPolicyAssignmentNameRequest:= "<organization_policy_assignment_name>"
        request.OrganizationPolicyAssignmentName = &organizationPolicyAssignmentNameRequest
        limitRequest:= int32(<limit>)
        request.Limit = &limitRequest
        markerRequest:= "<marker>"
        request.Marker = &markerRequest
    }
    response, err := client.ListOrganizationPolicyAssignments(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.18 查询指定组织合规规则

### 功能介绍

查询指定组织合规规则。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/organizations/{organization\_id}/policy-assignments/  
{organization\_policy\_assignment\_id}

表 4-320 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34
organization_policy_assignment_id	是	String	组织合规规则ID。 最大长度：32

## 请求参数

表 4-321 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 200

表 4-322 响应 Body 参数

参数	参数类型	描述
owner_id	String	组织合规规则创建者。
organization_id	String	组织ID。
organization_policy_assignment_urn	String	组织合规规则资源唯一标识。
organization_policy_assignment_id	String	组织合规规则ID。
organization_policy_assignment_name	String	组织合规规则名称。
excluded_accounts	Array of strings	排除配置规则的账号。
description	String	描述信息。
period	String	触发周期。
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
parameters	Map<String, <b>PolicyParameter</b> Value>	规则参数。
policy_definition_id	String	策略ID。

参数	参数类型	描述
created_at	String	创建时间。
updated_at	String	更新时间。

表 4-323 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度：128
resource_provider	String	云服务名称。 最大长度：128
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-324 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

状态码：400

表 4-325 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-326 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-327 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "owner_id": "e74e043fab784a45ad88f5ef6a4bcffc",
  "organization_id": "o-doxbpkzqsdd51ti1k27w2y8fitakrkn",
  "organization_policy_assignment_urn":
  "rms::e74e043fab784a45ad88f5ef6a4bcffc:organizationPolicyAssignment:o-
  doxbpkzqsdd51ti1k27w2y8fitakrkn/d02b7fa9d5a74e638c1402d0868f71fd",
  "organization_policy_assignment_id": "d02b7fa9d5a74e638c1402d0868f71fd",
  "organization_policy_assignment_name": "allowed-images-by-id",
  "description": "The ECS resource is non-compliant if the image it used is not in the allowed list.",
  "period": null,
  "policy_filter": {
    "region_id": null,
    "resource_provider": "evs",
    "resource_type": "volumes",
    "resource_id": null,
    "tag_key": null,
    "tag_value": null
  },
  "policy_definition_id": "5fa265c0aa1e6afc05a0ff07",
  "parameters": { },
  "created_at": "2022-01-18T06:37:54.432Z",
  "updated_at": "2022-01-18T06:37:54.432Z"
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowOrganizationPolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowOrganizationPolicyAssignmentRequest request = new
        ShowOrganizationPolicyAssignmentRequest();
        try {
            ShowOrganizationPolicyAssignmentResponse response =
            client.showOrganizationPolicyAssignment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

```
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = ShowOrganizationPolicyAssignmentRequest()
    response = client.show_organization_policy_assignment(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.5.19 删除组织合规规则

#### 功能介绍

删除组织合规规则。

#### 调用方法

请参见[如何调用API](#)。

#### URI

DELETE /v1/resource-manager/organizations/{organization\_id}/policy-assignments/{organization\_policy\_assignment\_id}

表 4-328 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34
organization_policy_assignment_id	是	String	组织合规规则ID。 最大长度：32

## 请求参数

表 4-329 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 400

表 4-330 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-331 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-332 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeleteOrganizationPolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteOrganizationPolicyAssignmentRequest request = new
        DeleteOrganizationPolicyAssignmentRequest();
        try {
            DeleteOrganizationPolicyAssignmentResponse response =
            client.deleteOrganizationPolicyAssignment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
```

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

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeleteOrganizationPolicyAssignmentRequest()
        response = client.delete_organization_policy_assignment(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
204	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.20 更新组织合规规则

### 功能介绍

更新组织合规规则

### 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1/resource-manager/organizations/{organization\_id}/policy-assignments/{organization\_policy\_assignment\_id}

表 4-333 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34
organization_policy_assignment_id	是	String	组织合规规则ID。 最大长度：32

## 请求参数

表 4-334 请求 Body 参数

参数	是否必选	参数类型	描述
excluded_accounts	否	Array of strings	需要排除配置规则的账号。 最大长度：32
organization_policy_assignment_name	是	String	组织合规规则名称。 最大长度：60
managed_policy_assignment_metadata	否	<b>ManagedPolicyAssignmentMetadata</b> object	托管规则元数据。

表 4-335 ManagedPolicyAssignmentMetadata

参数	是否必选	参数类型	描述
description	否	String	规则描述。 最大长度：512
period	否	String	触发周期。
parameters	否	Map<String, <b>PolicyParameterValue</b> >	输入参数。
policy_filter	否	<b>PolicyFilterDefinition</b> object	规则过滤器。
policy_definition_id	是	String	预定义策略标识符。 最大长度：36

表 4-336 PolicyParameterValue

参数	是否必选	参数类型	描述
value	否	Object	规则参数值。



表 4-337 PolicyFilterDefinition

参数	是否必选	参数类型	描述
region_id	否	String	区域ID。 最大长度：128
resource_provider	否	String	云服务名称。 最大长度：128
resource_type	否	String	资源类型。 最大长度：128
resource_id	否	String	资源ID。 最大长度：512
tag_key	否	String	标签键。 最大长度：128
tag_value	否	String	标签值。 最大长度：256

## 响应参数

状态码：200

表 4-338 响应 Body 参数

参数	参数类型	描述
owner_id	String	组织合规规则创建者。
organization_id	String	组织ID。
organization_policy_assignment_urn	String	组织合规规则资源唯一标识。
organization_policy_assignment_id	String	组织合规规则ID。
organization_policy_assignment_name	String	组织合规规则名称。
description	String	描述信息。
period	String	触发周期。

参数	参数类型	描述
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
parameters	Map<String, <b>PolicyParameter</b> <b>Value</b> >	规则参数。
policy_definition_id	String	策略ID。
created_at	String	创建时间。
updated_at	String	更新时间。

表 4-339 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度：128
resource_provider	String	云服务名称。 最大长度：128
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-340 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

状态码：400

表 4-341 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-342 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-343 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

更新组织合规规则。

```
PUT https://{endpoint}/v1/resource-manager/organizations/{organization_id}/policy-assignments/{organization_policy_assignment_id}
{
  "organization_policy_assignment_name": "allowed-images-by-id",
  "managed_policy_assignment_metadata": {
    "description": "The ECS resource is non-compliant if the image it used is not in the allowed list.",
    "parameters": {
      "listOfAllowedImages": {
        "value": [ "ea0d6e0e-99c3-406d-a873-3bb45462b624" ]
      }
    },
    "policy_filter": {
      "resource_provider": "ecs",
      "resource_type": "cloudservers"
    },
    "policy_definition_id": "5fa265c0aa1e6afc05a0ff07"
  }
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

更新组织合规规则。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateOrganizationPolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateOrganizationPolicyAssignmentRequest request = new
UpdateOrganizationPolicyAssignmentRequest();
        OrganizationPolicyAssignmentRequest body = new OrganizationPolicyAssignmentRequest();
        PolicyFilterDefinition policyFilterManagedPolicyAssignmentMetadata = new PolicyFilterDefinition();
        policyFilterManagedPolicyAssignmentMetadata.withResourceProvider("ecs")
            .withResourceType("cloudservers");
        PolicyParameterValue parametersParameters = new PolicyParameterValue();
        parametersParameters.withValue("[ea0d6e0e-99c3-406d-a873-3bb45462b624]");
        Map<String, PolicyParameterValue> listManagedPolicyAssignmentMetadataParameters = new
HashMap<>();
        listManagedPolicyAssignmentMetadataParameters.put("listOfAllowedImages", parametersParameters);
        ManagedPolicyAssignmentMetadata managedPolicyAssignmentMetadatabody = new
ManagedPolicyAssignmentMetadata();
        managedPolicyAssignmentMetadatabody.withDescription("The ECS resource is non-compliant if the
image it used is not in the allowed list.")
            .withParameters(listManagedPolicyAssignmentMetadataParameters)
            .withPolicyFilter(policyFilterManagedPolicyAssignmentMetadata)
            .withPolicyDefinitionId("5fa265c0aa1e6afc05a0ff07");
        body.withManagedPolicyAssignmentMetadata(managedPolicyAssignmentMetadatabody);
        body.withOrganizationPolicyAssignmentName("allowed-images-by-id");
        request.withBody(body);
        try {
```

```
        UpdateOrganizationPolicyAssignmentResponse response =
client.updateOrganizationPolicyAssignment(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

更新组织合规规则。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = UpdateOrganizationPolicyAssignmentRequest()
        policyFilterManagedPolicyAssignmentMetadata = PolicyFilterDefinition(
            resource_provider="ecs",
            resource_type="cloudservers"
        )
        parametersParameters = PolicyParameterValue(
            value="[ea0d6e0e-99c3-406d-a873-3bb45462b624]"
        )
        listParametersManagedPolicyAssignmentMetadata = {
            "listOfAllowedImages": parametersParameters
        }
        managedPolicyAssignmentMetadatabody = ManagedPolicyAssignmentMetadata(
            description="The ECS resource is non-compliant if the image it used is not in the allowed list.",
            parameters=listParametersManagedPolicyAssignmentMetadata,
            policy_filter=policyFilterManagedPolicyAssignmentMetadata,
            policy_definition_id="5fa265c0aa1e6afc05a0ff07"
        )
        request.body = OrganizationPolicyAssignmentRequest(
            managed_policy_assignment_metadata=managedPolicyAssignmentMetadatabody,
            organization_policy_assignment_name="allowed-images-by-id"
        )
        response = client.update_organization_policy_assignment(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

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

## Go

更新组织合规规则。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateOrganizationPolicyAssignmentRequest{
        resourceProviderPolicyFilter:= "ecs"
        resourceTypePolicyFilter:= "cloudservers"
        policyFilterManagedPolicyAssignmentMetadata := &model.PolicyFilterDefinition{
            ResourceProvider: &resourceProviderPolicyFilter,
            ResourceType: &resourceTypePolicyFilter,
        }
        var valueParameters interface{} = "[ea0d6e0e-99c3-406d-a873-3bb45462b624]"
        parametersParameters := model.PolicyParameterValue{
            Value: &valueParameters,
        }
        var listParametersManagedPolicyAssignmentMetadata = map[string](model.PolicyParameterValue){
            "listOfAllowedImages": parametersParameters,
        }
        descriptionManagedPolicyAssignmentMetadata:= "The ECS resource is non-compliant if the image it
        used is not in the allowed list."
        managedPolicyAssignmentMetadatabody := &model.ManagedPolicyAssignmentMetadata{
            Description: &descriptionManagedPolicyAssignmentMetadata,
            Parameters: listParametersManagedPolicyAssignmentMetadata,
            PolicyFilter: policyFilterManagedPolicyAssignmentMetadata,
            PolicyDefinitionId: "5fa265c0aa1e6afc05a0ff07",
        }
        request.Body = &model.OrganizationPolicyAssignmentRequest{
            ManagedPolicyAssignmentMetadata: managedPolicyAssignmentMetadatabody,
            OrganizationPolicyAssignmentName: "allowed-images-by-id",
        }
        response, err := client.UpdateOrganizationPolicyAssignment(request)
        if err == nil {
            fmt.Printf("%v\n", response)
        } else {
```

```
    fmt.Println(err)
  }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功
400	输入参数不合法
403	用户认证失败或无操作权限
500	服务器内部错误

## 错误码

请参见[错误码](#)。

## 4.5.21 查询组织合规规则部署状态

### 功能介绍

查询组织合规规则部署状态。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/organizations/{organization\_id}/policy-assignment-statuses

表 4-344 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

表 4-345 Query 参数

参数	是否必选	参数类型	描述
organization_policy_assignment_id	否	String	组织合规规则ID 最大长度：512
organization_policy_assignment_name	否	String	组织合规规则名称。 最大长度：60
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-346 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-347 响应 Body 参数

参数	参数类型	描述
value	Array of <b>OrganizationPolicyAssignmentStatusResponse</b> objects	组织合规规则部署状态结果列表。



参数	参数类型	描述
page_info	<b>PageInfo</b> object	分页对象。

表 4-348 OrganizationPolicyAssignmentStatusResponse

参数	参数类型	描述
organization_policy_assignment_id	String	组织合规规则ID。
organization_policy_assignment_name	String	组织合规规则名称。
organization_policy_assignment_status	String	组织合规规则部署状态。
error_code	String	当创建或更新组织合规规则失败时错误码。
error_message	String	当创建或更新组织合规规则失败时错误信息。
created_at	String	创建时间。
updated_at	String	更新时间。

表 4-349 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-350 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-351 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-352 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [{
    "organization_policy_assignment_id": "9642977cef864971b3b00ad56621a15f",
    "organization_policy_assignment_name": "allowed-images-by-id",
    "organization_policy_assignment_status": "CREATE_SUCCESSFUL",
    "error_code": "null",
    "error_message": "null",
    "created_at": "2023-01-17T09:38:26.165Z",
    "updated_at": "2023-01-17T09:38:26.165Z"
  }],
  "page_info": {
    "current_count": 1,
    "next_marker": "CAESIGQwMml3ZmE5ZDVhNzRlNjM4YzE0MDJkMDg2OGY3MWZkGgTnpS5u"
  }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowOrganizationPolicyAssignmentStatusesSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowOrganizationPolicyAssignmentStatusesRequest request = new
        ShowOrganizationPolicyAssignmentStatusesRequest();
        request.withOrganizationPolicyAssignmentId("<organization_policy_assignment_id>");
        request.withOrganizationPolicyAssignmentName("<organization_policy_assignment_name>");
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        try {
            ShowOrganizationPolicyAssignmentStatusesResponse response =
            client.showOrganizationPolicyAssignmentStatuses(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowOrganizationPolicyAssignmentStatusesRequest()
        request.organization_policy_assignment_id = "<organization_policy_assignment_id>"
        request.organization_policy_assignment_name = "<organization_policy_assignment_name>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.show_organization_policy_assignment_statuses(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowOrganizationPolicyAssignmentStatusesRequest{
        organizationPolicyAssignmentIdRequest:= "<organization_policy_assignment_id>"
        request.OrganizationPolicyAssignmentId = &organizationPolicyAssignmentIdRequest
        organizationPolicyAssignmentNameRequest:= "<organization_policy_assignment_name>"
        request.OrganizationPolicyAssignmentName = &organizationPolicyAssignmentNameRequest
    }
```

```
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
response, err := client.ShowOrganizationPolicyAssignmentStatuses(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.5.22 查询组织内每个成员账号合规规则部署的详细状态

### 功能介绍

查询组织内每个成员账号合规规则部署的详细状态。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/organizations/{organization\_id}/policy-assignment-detailed-status

表 4-353 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

表 4-354 Query 参数

参数	是否必选	参数类型	描述
organization_policy_assignment_name	是	String	组织合规规则名称。 最大长度：60
organization_policy_assignment_id	否	String	组织合规规则ID 最大长度：512
domain_id	否	String	账号ID。 最大长度：36
status	否	String	成员账号规则部署状态，区分大小写。 最大长度：20
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-355 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-356 响应 Body 参数

参数	参数类型	描述
value	Array of <b>OrganizationPolicyAssignmentDetailedStatusResponse</b> objects	组织合规规则部署详细状态结果列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-357 OrganizationPolicyAssignmentDetailedStatusResponse

参数	参数类型	描述
domain_id	String	账号ID。
policy_assignment_id	String	合规规则ID。
policy_assignment_name	String	合规规则名称。
member_account_policy_assignment_status	String	成员账号中配置规则的部署状态。
error_code	String	当创建或更新合规规则失败时错误码。
error_message	String	当创建或更新合规规则失败时错误信息。
created_at	String	创建时间。
updated_at	String	更新时间。

表 4-358 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-359 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-360 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-361 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [ {
    "domain_id": "3fd23817bd85480e8dc4f520075de348",
    "policy_assignment_id": "63c79416e34d812e56d3e2fc",
    "policy_assignment_name": "allowed-images-by-id",
    "member_account_policy_assignment_status": "CREATE_SUCCESSFUL",
    "error_code": "null",
    "error_message": "null",
    "created_at": "2023-01-17T09:38:26.165Z",
    "updated_at": "2023-01-17T09:38:26.165Z"
  } ],
  "page_info": {
    "current_count": 1,

```



```
"next_marker" : "CAESIGQwMml3ZmE5ZDVhNzRlNjM4YzE0MDJkMDg2OGY3MWZkGgTnpS5u"  
}  
}
```

## 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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class ShowOrganizationPolicyAssignmentDetailedStatusSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowOrganizationPolicyAssignmentDetailedStatusRequest request = new  
        ShowOrganizationPolicyAssignmentDetailedStatusRequest();  
        request.withOrganizationPolicyAssignmentName("<organization_policy_assignment_name>");  
        request.withOrganizationPolicyAssignmentId("<organization_policy_assignment_id>");  
  
        request.withStatus(ShowOrganizationPolicyAssignmentDetailedStatusRequest.StatusEnum.fromValue("<stat  
us>"));  
        request.withLimit(<limit>);  
        request.withMarker("<marker>");  
        try {  
            ShowOrganizationPolicyAssignmentDetailedStatusResponse response =  
            client.showOrganizationPolicyAssignmentDetailedStatus(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowOrganizationPolicyAssignmentDetailedStatusRequest()
        request.organization_policy_assignment_name = "<organization_policy_assignment_name>"
        request.organization_policy_assignment_id = "<organization_policy_assignment_id>"
        request.status = "<status>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.show_organization_policy_assignment_detailed_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).
Build()

request := &model.ShowOrganizationPolicyAssignmentDetailedStatusRequest{}
request.OrganizationPolicyAssignmentName = "<organization_policy_assignment_name>"
organizationPolicyAssignmentIdRequest:= "<organization_policy_assignment_id>"
request.OrganizationPolicyAssignmentId = &organizationPolicyAssignmentIdRequest
statusRequest:=
model.GetShowOrganizationPolicyAssignmentDetailedStatusRequestStatusEnum().<STATUS>
request.Status = &statusRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
response, err := client.ShowOrganizationPolicyAssignmentDetailedStatus(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.6 区域管理

### 4.6.1 查询用户可见的区域

#### 功能介绍

查询用户可见的区域。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/regions

表 4-362 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

## 请求参数

表 4-363 请求 Header 参数

参数	是否必选	参数类型	描述
X-Language	否	String	选择接口返回的信息的语言，默认为"zh-cn"中文，可选值：zh-cn和en-us。 缺省值： <b>zh-cn</b>
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 200

表 4-364 响应 Body 参数

参数	参数类型	描述
value	Array of <a href="#">Region</a> objects	区域信息项列表。

表 4-365 Region

参数	参数类型	描述
region_id	String	区域ID。
display_name	String	显示名称。

状态码： 400

表 4-366 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-367 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-368 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-369 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

**状态码： 200**

操作成功。

```
{
  "value": [{
    "region_id": "regionid1",
    "display_name": "区域一"
  }, {
    "region_id": "regionid2",
    "display_name": "区域二"
  }]
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListRegionsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListRegionsRequest request = new ListRegionsRequest();
        try {
            ListRegionsResponse response = client.listRegions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkconfig.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = GlobalCredentials(ak, sk) \  
  
    client = ConfigClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListRegionsRequest()  
        response = client.list_regions(request)  
        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/global"  
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).
```

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
404	未找到用户。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.7 高级查询

### 4.7.1 运行高级查询

#### 功能介绍

执行高级查询。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/resource-manager/domains/{domain\_id}/run-query



表 4-370 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

## 请求参数

表 4-371 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-372 请求 Body 参数

参数	是否必选	参数类型	描述
expression	是	String	ResourceQL表达式。 最小长度：1 最大长度：4096

## 响应参数

状态码：200

表 4-373 响应 Body 参数

参数	参数类型	描述
query_info	<b>QueryInfo</b> object	ResourceQL查询字段。
results	Array of objects	ResourceQL查询结果。

表 4-374 QueryInfo

参数	参数类型	描述
select_fields	Array of strings	ResourceQL查询字段。

状态码： 400

表 4-375 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-376 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 429

表 4-377 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-378 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

- 查询当前用户下所有虚拟机的id。

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/run-query
{
  "expression": "select id from resources where provider = 'ecs' and type = 'cloudservers'"
}
```

- 查询当前用户下指定容量为100GB的云硬盘。

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/run-query
{
  "expression": "select * from resources where provider = 'evs' and type = 'volumes' and properties.size = 100"
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "query_info": {
    "select_fields": [ "id" ]
  },
  "results": [ {
    "id": "91252cc9-bfd9-0709-0912-56b397e0ba3f"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 查询当前用户下所有虚拟机的id。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class RunQuerySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
```

```
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    RunQueryRequest request = new RunQueryRequest();
    QueryRunRequestBody body = new QueryRunRequestBody();
    body.withExpression("select id from resources where provider = 'ecs' and type = 'cloudservers'");
    request.withBody(body);
    try {
        RunQueryResponse response = client.runQuery(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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());
    }
}
}
```

- 查询当前用户下指定容量为100GB的云硬盘。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class RunQuerySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        RunQueryRequest request = new RunQueryRequest();
        QueryRunRequestBody body = new QueryRunRequestBody();
        body.withExpression("select * from resources where provider = 'evs' and type = 'volumes' and
properties.size = 100");
        request.withBody(body);
        try {
            RunQueryResponse response = client.runQuery(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

- 查询当前用户下所有虚拟机的id。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = RunQueryRequest()
        request.body = QueryRunRequestBody(
            expression="select id from resources where provider = 'ecs' and type = 'cloudservers'"
        )
        response = client.run_query(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 查询当前用户下指定容量为100GB的云硬盘。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

```
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = RunQueryRequest()
    request.body = QueryRunRequestBody(
        expression="select * from resources where provider = 'evs' and type = 'volumes' and
properties.size = 100"
    )
    response = client.run_query(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

- 查询当前用户下所有虚拟机的id。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RunQueryRequest{}
    request.Body = &model.QueryRunRequestBody{
        Expression: "select id from resources where provider = 'ecs' and type = 'cloudservers'",
    }
    response, err := client.RunQuery(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 查询当前用户下指定容量为100GB的云硬盘。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RunQueryRequest{}
    request.Body = &model.QueryRunRequestBody{
        Expression: "select * from resources where provider = 'evs' and type = 'volumes' and
        properties.size = 100",
    }
    response, err := client.RunQuery(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
429	超出频次限制。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.7.2 创建高级查询

### 功能介绍

创建新的高级查询。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/domains/{domain\_id}/stored-queries

表 4-379 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

### 请求参数

表 4-380 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-381 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	ResourceQL名称。 最小长度：1 最大长度：64



参数	是否必选	参数类型	描述
description	否	String	ResourceQL描述。 最小长度：0 最大长度：512
expression	是	String	ResourceQL表达式。 最小长度：1 最大长度：4096

## 响应参数

状态码：200

表 4-382 响应 Body 参数

参数	参数类型	描述
id	String	ResourceQL ID。 最大长度：256
name	String	ResourceQL名称。 最小长度：1 最大长度：64
description	String	ResourceQL描述。 最小长度：0 最大长度：512
expression	String	ResourceQL表达式。 最小长度：1 最大长度：4096
created	String	ResourceQL创建时间。 最小长度：1 最大长度：64
updated	String	ResourceQL更新时间。 最小长度：1 最大长度：64

状态码：400

表 4-383 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-384 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-385 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

创建高级查询，用于查询关机状态的云服务器。

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/stored-queries
{
  "name": "stopped-ecs",
  "description": "查询关机状态的云服务器。",
  "expression": "SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers' AND properties.status = 'SHUTOFF'"
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "id": "62b9126566d400721efceffa",
  "name": "stopped-ecs",
  "description": "查询关机状态的云服务器。",
  "expression": "SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers' AND properties.status = 'SHUTOFF'",
  "created": "2022-06-27T02:13:57.107Z",
}
```

```
"updated" : "2022-06-27T02:13:57.107Z"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建高级查询，用于查询关机状态的云服务器。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class CreateStoredQuerySolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateStoredQueryRequest request = new CreateStoredQueryRequest();  
        StoredQueryRequestBody body = new StoredQueryRequestBody();  
        body.withExpression("SELECT id, name FROM resources WHERE provider = 'ecs' AND type =  
'cloudservers' AND properties.status = 'SHUTOFF'");  
        body.withDescription("查询关机状态的云服务器。");  
        body.withName("stopped-ecs");  
        request.withBody(body);  
        try {  
            CreateStoredQueryResponse response = client.createStoredQuery(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } 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

创建高级查询，用于查询关机状态的云服务器。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CreateStoredQueryRequest()
        request.body = StoredQueryRequestBody(
            expression="SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers'
            AND properties.status = 'SHUTOFF",
            description="查询关机状态的云服务器。",
            name="stopped-ecs"
        )
        response = client.create_stored_query(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

创建高级查询，用于查询关机状态的云服务器。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
```

```
WithSk(sk).
Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateStoredQueryRequest{
    descriptionStoredQueryRequestBody:= "查询关机状态的云服务器。"
    request.Body = &model.StoredQueryRequestBody{
        Expression: "SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers' AND
properties.status = 'SHUTOFF'",
        Description: &descriptionStoredQueryRequestBody,
        Name: "stopped-ecs",
    }
}
response, err := client.CreateStoredQuery(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.7.3 列举高级查询

#### 功能介绍

列举所有高级查询。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/stored-queries

表 4-386 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-387 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
name	否	String	ResourceQL名称。 最小长度：1 最大长度：64

## 请求参数

表 4-388 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-389 响应 Body 参数

参数	参数类型	描述
value	Array of <a href="#">StoredQuery</a> objects	高级查询列表。
page_info	<a href="#">PageInfo</a> object	分页对象。

表 4-390 StoredQuery

参数	参数类型	描述
id	String	ResourceQL ID。 最大长度：256
name	String	ResourceQL名称。 最小长度：1 最大长度：64
description	String	ResourceQL描述。 最小长度：0 最大长度：512
expression	String	ResourceQL表达式。 最小长度：1 最大长度：4096
created	String	ResourceQL创建时间。 最小长度：1 最大长度：64
updated	String	ResourceQL更新时间。 最小长度：1 最大长度：64

表 4-391 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200

参数	参数类型	描述
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-392 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-393 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-394 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码：200

操作成功。

```
{  
  "value": [ {
```



```
"id" : "62b9126566d400721efceffa",
"name" : "stopped-ecs",
"description" : "查询关机状态的云服务器。",
"expression" : "SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers' AND
properties.status = 'SHUTOFF'",
"created" : "2022-06-27T02:13:57.107Z",
"updated" : "2022-06-27T02:13:57.107Z"
}],
"page_info" : {
  "current_count" : 1,
  "next_marker" : 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListStoredQueriesSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListStoredQueriesRequest request = new ListStoredQueriesRequest();
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        request.withName("<name>");
        try {
            ListStoredQueriesResponse response = client.listStoredQueries(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkconfig.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = GlobalCredentials(ak, sk) \  
  
    client = ConfigClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListStoredQueriesRequest()  
        request.limit = <limit>  
        request.marker = "<marker>"  
        request.name = "<name>"  
        response = client.list_stored_queries(request)  
        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/global"  
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  

```

```
config.ConfigClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build())

request := &model.ListStoredQueriesRequest{}
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
nameRequest:= "<name>"
request.Name = &nameRequest
response, err := client.ListStoredQueries(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.7.4 查询高级查询

### 功能介绍

查询单个高级查询。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/stored-queries/{query\_id}

表 4-395 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
query_id	是	String	查询ID。 最大长度：36

## 请求参数

表 4-396 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-397 响应 Body 参数

参数	参数类型	描述
id	String	ResourceQL ID。 最大长度：256
name	String	ResourceQL名称。 最小长度：1 最大长度：64
description	String	ResourceQL描述。 最小长度：0 最大长度：512
expression	String	ResourceQL表达式。 最小长度：1 最大长度：4096

参数	参数类型	描述
created	String	ResourceQL创建时间。 最小长度：1 最大长度：64
updated	String	ResourceQL更新时间。 最小长度：1 最大长度：64

状态码：400

表 4-398 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-399 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-400 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-401 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

**状态码： 200**

操作成功。

```
{
  "id": "62b9126566d400721efceffa",
  "name": "stopped-ecs",
  "description": "查询关机状态的云服务器。",
  "expression": "SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers' AND properties.status = 'SHUTOFF'",
  "created": "2022-06-27T02:13:57.107Z",
  "updated": "2022-06-27T02:13:57.107Z"
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowStoredQuerySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
ShowStoredQueryRequest request = new ShowStoredQueryRequest();
try {
    ShowStoredQueryResponse response = client.showStoredQuery(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowStoredQueryRequest()
        response = client.show_stored_query(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

func main() {
```

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

auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
404	未找到资源。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.7.5 更新单个高级查询

### 功能介绍

更新单个自定义查询。



## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1/resource-manager/domains/{domain\_id}/stored-queries/{query\_id}

表 4-402 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
query_id	是	String	查询ID。 最大长度：36

## 请求参数

表 4-403 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-404 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	ResourceQL名称。 最小长度：1 最大长度：64
description	否	String	ResourceQL描述。 最小长度：0 最大长度：512
expression	是	String	ResourceQL表达式。 最小长度：1 最大长度：4096

## 响应参数

状态码： 200

表 4-405 响应 Body 参数

参数	参数类型	描述
id	String	ResourceQL ID。 最大长度： 256
name	String	ResourceQL名称。 最小长度： 1 最大长度： 64
description	String	ResourceQL描述。 最小长度： 0 最大长度： 512
expression	String	ResourceQL表达式。 最小长度： 1 最大长度： 4096
created	String	ResourceQL创建时间。 最小长度： 1 最大长度： 64
updated	String	ResourceQL更新时间。 最小长度： 1 最大长度： 64

状态码： 400

表 4-406 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-407 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	错误消息内容。

状态码： 404

表 4-408 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-409 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

更新高级查询，用于查询开机状态的云服务器。

```
PUT https://{endpoint}/v1/resource-manager/domains/{domain_id}/stored-queries/{query_id}
```

```
{
  "name": "running-ecs",
  "description": "查询开机状态的云服务器。",
  "expression": "SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers' AND properties.status = 'ACTIVE'"
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "id": "62b9126566d400721efceffa",
  "name": "running-ecs",
  "description": "查询开机状态的云服务器。",
  "expression": "SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers' AND properties.status = 'ACTIVE'",
  "created": "2022-06-27T02:13:57.107Z",
  "updated": "2022-07-09T09:12:00.007Z"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新高级查询，用于查询开机状态的云服务器。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class UpdateStoredQuerySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateStoredQueryRequest request = new UpdateStoredQueryRequest();
        StoredQueryRequestBody body = new StoredQueryRequestBody();
        body.withExpression("SELECT id, name FROM resources WHERE provider = 'ecs' AND type =
'cloudservers' AND properties.status = 'ACTIVE'");
        body.withDescription("查询开机状态的云服务器。");
        body.withName("running-ecs");
        request.withBody(body);
        try {
            UpdateStoredQueryResponse response = client.updateStoredQuery(request);
            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

更新高级查询，用于查询开机状态的云服务器。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = UpdateStoredQueryRequest()
        request.body = StoredQueryRequestBody(
            expression="SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers'
            AND properties.status = 'ACTIVE'",
            description="查询开机状态的云服务器。",
            name="running-ecs"
        )
        response = client.update_stored_query(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

更新高级查询，用于查询开机状态的云服务器。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
```

```
WithRegion(region.ValueOf("<YOUR REGION>")).
WithCredential(auth).
Build()

request := &model.UpdateStoredQueryRequest{
descriptionStoredQueryRequestBody:= "查询开机状态的云服务器。"
request.Body = &model.StoredQueryRequestBody{
Expression: "SELECT id, name FROM resources WHERE provider = 'ecs' AND type = 'cloudservers' AND
properties.status = 'ACTIVE'",
Description: &descriptionStoredQueryRequestBody,
Name: "running-ecs",
}
response, err := client.UpdateStoredQuery(request)
if err == nil {
fmt.Printf("%+v\n", response)
} else {
fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败。
404	未找到资源。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.7.6 删除高级查询

### 功能介绍

删除单个高级查询。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/resource-manager/domains/{domain\_id}/stored-queries/{query\_id}

表 4-410 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
query_id	是	String	查询ID。 最大长度：36

## 请求参数

表 4-411 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：400

表 4-412 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-413 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-414 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-415 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeleteStoredQuerySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
```



```
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    DeleteStoredQueryRequest request = new DeleteStoredQueryRequest();
    try {
        DeleteStoredQueryResponse response = client.deleteStoredQuery(request);
        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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeleteStoredQueryRequest()
        response = client.delete_stored_query(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
```

```
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.DeleteStoredQueryRequest{}  
    response, err := client.DeleteStoredQuery(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
204	操作成功。
400	输入参数不合法。
403	用户认证失败。
404	未找到资源。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.7.7 列举高级查询 Schema

#### 功能介绍

列举高级查询使用的Schema。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/schemas

表 4-416 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-417 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-418 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	调用者token。
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-419 响应 Body 参数

参数	参数类型	描述
value	Array of ResourceSchemaResponse objects	schemas接口。
page_info	PageInfo object	分页对象。

表 4-420 ResourceSchemaResponse

参数	参数类型	描述
type	String	资源类型。
schema	Object	schema 内容。

表 4-421 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-422 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-423 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-424 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [ {
    "type": "aad.instances",
    "schema": {
      "expireTime": "int",
      "ispSpec": "string",
      "specType": "int",
      "basicBandwidth": "int",
      "elasticBandwidth": "int",
      "serviceBandwidth": "int"
    }
  }
],
  "page_info": {
    "current_count": 1,
    "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.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");

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSchemasRequest request = new ListSchemasRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

```
.build()

try:
    request = ListSchemasRequest()
    request.limit = <limit>
    request.marker = "<marker>"
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSchemasRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    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	操作成功。
400	输入参数不合法。
403	用户认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8 资源聚合器

### 4.8.1 创建资源聚合器

#### 功能介绍

创建资源聚合器。

#### 调用方法

请参见[如何调用API](#)。

#### URI

PUT /v1/resource-manager/domains/{domain\_id}/aggregators

表 4-425 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36



## 请求参数

表 4-426 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-427 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_name	是	String	资源聚合器名称。 最小长度：1 最大长度：64
aggregator_type	是	String	聚合器类型（ACCOUNT   ORGANIZATION）。
account_aggregation_sources	否	AccountAggregationSource object	聚合数据的源账号。

表 4-428 AccountAggregationSource

参数	是否必选	参数类型	描述
domain_ids	否	Array of strings	账号列表。 最小长度：1 最大长度：36 数组长度：1 - 30

## 响应参数

状态码：200

表 4-429 响应 Body 参数

参数	参数类型	描述
aggregator_name	String	资源聚合器名称。
aggregator_id	String	资源聚合器ID。

参数	参数类型	描述
aggregator_url	String	资源聚合器标识符。
aggregator_type	String	聚合器类型。
account_aggregation_sources	<a href="#">AccountAggregationSource</a> object	聚合数据的源账号。
updated_at	String	资源聚合器更新时间。
created_at	String	资源聚合器创建时间。

表 4-430 AccountAggregationSource

参数	参数类型	描述
domain_ids	Array of strings	账号列表。 最小长度：1 最大长度：36 数组长度：1 - 30

状态码：400

表 4-431 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-432 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-433 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

创建一个账号类型的资源聚合器。

```
PUT https://{endpoint}/v1/resource-manager/domains/{domain_id}/aggregators
{
  "aggregator_name": "example_aggregator",
  "aggregator_type": "ACCOUNT",
  "account_aggregation_sources": {
    "domain_ids": [ "3fd23817bd85480e8dc4f520075de348" ]
  }
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "aggregator_name": "example_aggregator",
  "aggregator_id": "7ca3447830bd49c9b630a99680eaf90f",
  "aggregator_urn":
  "rms::e74e043fab784a45ad88f5ef6a4bcffc:aggregationAggregator:7ca3447830bd49c9b630a99680eaf90f",
  "aggregator_type": "ACCOUNT",
  "account_aggregation_sources": {
    "domain_ids": [ "3fd23817bd85480e8dc4f520075de348" ]
  },
  "updated_at": "2020-01-18T02:49:12.465Z",
  "created_at": "2020-01-18T02:49:12.465Z"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个账号类型的资源聚合器。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

import java.util.List;
```

```
import java.util.ArrayList;

public class CreateConfigurationAggregatorSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateConfigurationAggregatorRequest request = new CreateConfigurationAggregatorRequest();
        ConfigurationAggregatorRequest body = new ConfigurationAggregatorRequest();
        List<String> listAccountAggregationSourcesDomainIds = new ArrayList<>();
        listAccountAggregationSourcesDomainIds.add("3fd23817bd85480e8dc4f520075de348");
        AccountAggregationSource accountAggregationSourcesbody = new AccountAggregationSource();
        accountAggregationSourcesbody.withDomainIds(listAccountAggregationSourcesDomainIds);
        body.withAccountAggregationSources(accountAggregationSourcesbody);
        body.withAggregatorType("ACCOUNT");
        body.withAggregatorName("example_aggregator");
        request.withBody(body);
        try {
            CreateConfigurationAggregatorResponse response = client.createConfigurationAggregator(request);
            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

创建一个账号类型的资源聚合器。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \
```

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

try:
    request = CreateConfigurationAggregatorRequest()
    listDomainIdsAccountAggregationSources = [
        "3fd23817bd85480e8dc4f520075de348"
    ]
    accountAggregationSourcesbody = AccountAggregationSource(
        domain_ids=listDomainIdsAccountAggregationSources
    )
    request.body = ConfigurationAggregatorRequest(
        account_aggregation_sources=accountAggregationSourcesbody,
        aggregator_type="ACCOUNT",
        aggregator_name="example_aggregator"
    )
    response = client.create_configuration_aggregator(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

创建一个账号类型的资源聚合器。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConfigurationAggregatorRequest{
        var listDomainIdsAccountAggregationSources = []string{
            "3fd23817bd85480e8dc4f520075de348",
        }
    }
    accountAggregationSourcesbody := &model.AccountAggregationSource{
        DomainIds: &listDomainIdsAccountAggregationSources,
    }
    request.Body = &model.ConfigurationAggregatorRequest{
```

```
AccountAggregationSources: accountAggregationSourcesbody,  
AggregatorType: "ACCOUNT",  
AggregatorName: "example_aggregator",  
}  
response, err := client.CreateConfigurationAggregator(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.2 查询资源聚合器列表

### 功能介绍

查询资源聚合器列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/aggregators

表 4-434 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-435 Query 参数

参数	是否必选	参数类型	描述
aggregator_name	否	String	资源聚合器名称。 最小长度：1 最大长度：64
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-436 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-437 响应 Body 参数

参数	参数类型	描述
configuration_aggregators	Array of <b>ConfigurationAggregatorResp</b> objects	资源聚合器列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-438 ConfigurationAggregatorResp

参数	参数类型	描述
aggregator_name	String	资源聚合器名称。
aggregator_id	String	资源聚合器ID。
aggregator_url	String	资源聚合器标识符。
aggregator_type	String	聚合器类型。
account_aggregation_sources	<a href="#">AccountAggregationSource</a> object	聚合数据的源账号。
updated_at	String	资源聚合器更新时间。
created_at	String	资源聚合器创建时间。

表 4-439 AccountAggregationSource

参数	参数类型	描述
domain_ids	Array of strings	账号列表。 最小长度：1 最大长度：36 数组长度：1 - 30

表 4-440 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400



表 4-441 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-442 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-443 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "configuration_aggregators": [ {
    "aggregator_name": "example_aggregator",
    "aggregator_id": "2f0cc4991bac4940ae20a3630818589e",
    "aggregator_urn":
    "rms::e74e043fab784a45ad88f5ef6a4bcffc:aggregationAggregator:2f0cc4991bac4940ae20a3630818589e",
    "aggregator_type": "ACCOUNT",
    "account_aggregation_sources": {
      "domain_ids": [ "3fd23817bd85480e8dc4f520075de348" ]
    },
    "updated_at": "2020-01-18T02:49:12.465Z",
    "created_at": "2020-01-18T02:49:12.465Z"
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": "CAESIDdjYTM0NDc4MzBiZDQ5YzliNjMwYTk5NjgwZWFMOTBmGgQ-0aw3"
  }
}
```

```
}  
}
```

## 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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class ListConfigurationAggregatorsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListConfigurationAggregatorsRequest request = new ListConfigurationAggregatorsRequest();  
        request.withAggregatorName("<aggregator_name>");  
        request.withLimit(<limit>);  
        request.withMarker("<marker>");  
        try {  
            ListConfigurationAggregatorsResponse response = client.listConfigurationAggregators(request);  
            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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
```

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

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListConfigurationAggregatorsRequest()
        request.aggregator_name = "<aggregator_name>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.list_configuration_aggregators(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListConfigurationAggregatorsRequest{}
    aggregatorNameRequest := "<aggregator_name>"
    request.AggregatorName = &aggregatorNameRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
```

```
markerRequest:= "<marker>"
request.Marker = &markerRequest
response, err := client.ListConfigurationAggregators(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.3 查询指定资源聚合器

### 功能介绍

查询指定资源聚合器。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/aggregators/{aggregator\_id}

表 4-444 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID。 最大长度：32

## 请求参数

表 4-445 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-446 响应 Body 参数

参数	参数类型	描述
aggregator_name	String	资源聚合器名称。
aggregator_id	String	资源聚合器ID。
aggregator_url	String	资源聚合器标识符。
aggregator_type	String	聚合器类型。
account_aggregation_sources	<b>AccountAggregationSource</b> object	聚合数据的源账号。
updated_at	String	资源聚合器更新时间。
created_at	String	资源聚合器创建时间。

表 4-447 AccountAggregationSource

参数	参数类型	描述
domain_ids	Array of strings	账号列表。 最小长度：1 最大长度：36 数组长度：1 - 30

状态码：400

表 4-448 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-449 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-450 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码：200

操作成功。

```
{
  "aggregator_name" : "example_aggregator",
  "aggregator_id" : "2f0cc4991bac4940ae20a3630818589e",
  "aggregator_urn" :
  "rms::e74e043fab784a45ad88f5ef6a4bcffc:aggregationAggregator:2f0cc4991bac4940ae20a3630818589e",
  "aggregator_type" : "ACCOUNT",
  "account_aggregation_sources" : {
    "domain_ids" : [ "3fd23817bd85480e8dc4f520075de348" ]
  },
  "updated_at" : "2020-01-18T02:49:12.465Z",
  "created_at" : "2020-01-18T02:49:12.465Z"
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowConfigurationAggregatorSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowConfigurationAggregatorRequest request = new ShowConfigurationAggregatorRequest();
        try {
            ShowConfigurationAggregatorResponse response = client.showConfigurationAggregator(request);
            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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkconfig.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = GlobalCredentials(ak, sk) \  
  
    client = ConfigClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ShowConfigurationAggregatorRequest()  
        response = client.show_configuration_aggregator(request)  
        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/global"  
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())
```



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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.4 查询指定资源聚合器聚合账号的状态信息

### 功能介绍

查询指定资源聚合器聚合账号的状态信息，状态包括验证源账号和聚合器账号之间授权的信息。如果失败，状态包含相关的错误码或消息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/aggregators/{aggregator\_id}/aggregator-sources-status

表 4-451 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID。 最大长度：32

表 4-452 Query 参数

参数	是否必选	参数类型	描述
update_status	否	String	聚合账号的状态。 最大长度：16
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-453 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-454 响应 Body 参数

参数	参数类型	描述
aggregated_source_statuses	Array of <b>AggregatedSourceStatus</b> objects	资源聚合器状态列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-455 AggregatedSourceStatus

参数	参数类型	描述
last_error_code	String	源账号最近一次聚合失败时返回的错误码。
last_error_message	String	源账号最近一次聚合失败时返回的错误消息。
last_update_status	String	最近一次更新的状态类型。
last_update_time	String	最近一次更新的时间。
source_id	String	源账号ID或组织。
source_type	String	源账号类型 ( ACCOUNT   ORGANIZATION ) 。
source_name	String	账号名。

表 4-456 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码： 400

表 4-457 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-458 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-459 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "aggregated_source_statuses": [ {
    "last_error_code": null,
    "last_error_message": null,
    "last_update_status": "SUCCEEDED",
    "last_update_time": "2020-01-18T10:49:12.442+08:00",
    "source_id": "3fd23817bd85480e8dc4f520075de348",
    "source_name": "demo",
    "source_type": "ACCOUNT"
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": "CAESIDdjYTM0NDc4MzBiZDQ5YzliNjMwYTk5NjgwZWVmOTBmGgQ-0aw3"
  }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowConfigurationAggregatorSourcesStatusSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowConfigurationAggregatorSourcesStatusRequest request = new
        ShowConfigurationAggregatorSourcesStatusRequest();

        request.withUpdateStatus(ShowConfigurationAggregatorSourcesStatusRequest.UpdateStatusEnum.fromValue("<update_status>"));
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        try {
            ShowConfigurationAggregatorSourcesStatusResponse response =
            client.showConfigurationAggregatorSourcesStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
```

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

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowConfigurationAggregatorSourcesStatusRequest()
        request.update_status = "<update_status>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.show_configuration_aggregator_sources_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowConfigurationAggregatorSourcesStatusRequest{}
    updateStatusRequest:=
    model.GetShowConfigurationAggregatorSourcesStatusRequestUpdateStatusEnum().<UPDATE_STATUS>
    request.UpdateStatus = &updateStatusRequest
    limitRequest:= int32(<limit>)
```

```
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
response, err := client.ShowConfigurationAggregatorSourcesStatus(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.5 更新资源聚合器

### 功能介绍

更新资源聚合器。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v1/resource-manager/domains/{domain\_id}/aggregators/{aggregator\_id}

表 4-460 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID。 最大长度：32

## 请求参数

表 4-461 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-462 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_name	是	String	资源聚合器名称。 最小长度：1 最大长度：64
aggregator_type	是	String	聚合器类型（ACCOUNT   ORGANIZATION）。
account_aggregation_sources	否	AccountAggregationSource object	聚合数据的源账号。

表 4-463 AccountAggregationSource

参数	是否必选	参数类型	描述
domain_ids	否	Array of strings	账号列表。 最小长度：1 最大长度：36 数组长度：1 - 30

## 响应参数

状态码：200



表 4-464 响应 Body 参数

参数	参数类型	描述
aggregator_name	String	资源聚合器名称。
aggregator_id	String	资源聚合器ID。
aggregator_url	String	资源聚合器标识符。
aggregator_type	String	聚合器类型。
account_aggregation_sources	<a href="#">AccountAggregationSource</a> object	聚合数据的源账号。
updated_at	String	资源聚合器更新时间。
created_at	String	资源聚合器创建时间。

表 4-465 AccountAggregationSource

参数	参数类型	描述
domain_ids	Array of strings	账号列表。 最小长度：1 最大长度：36 数组长度：1 - 30

状态码：400

表 4-466 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-467 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	错误消息内容。

状态码： 500

表 4-468 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

更新资源聚合器。

```
PUT https://{endpoint}/v1/resource-manager/domains/{domain_id}/aggregators/{aggregator_id}
{
  "aggregator_name": "example_aggregator",
  "aggregator_type": "ACCOUNT",
  "account_aggregation_sources": {
    "domain_ids": [ "3fd23817bd85480e8dc4f520075de348" ]
  }
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "aggregator_name": "example_aggregator",
  "aggregator_id": "7ca3447830bd49c9b630a99680eaf90f",
  "aggregator_urn":
  "rms::e74e043fab784a45ad88f5ef6a4bcffc:aggregationAggregator:7ca3447830bd49c9b630a99680eaf90f",
  "aggregator_type": "ACCOUNT",
  "account_aggregation_sources": {
    "domain_ids": [ "3fd23817bd85480e8dc4f520075de348" ]
  },
  "updated_at": "2020-01-18T02:49:12.465Z",
  "created_at": "2020-01-18T02:49:12.465Z"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新资源聚合器。

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

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

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

public class UpdateConfigurationAggregatorSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

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

        UpdateConfigurationAggregatorRequest request = new UpdateConfigurationAggregatorRequest();
        ConfigurationAggregatorRequest body = new ConfigurationAggregatorRequest();
        List<String> listAccountAggregationSourcesDomainIds = new ArrayList<>();
        listAccountAggregationSourcesDomainIds.add("3fd23817bd85480e8dc4f520075de348");
        AccountAggregationSource accountAggregationSourcesbody = new AccountAggregationSource();
        accountAggregationSourcesbody.withDomainIds(listAccountAggregationSourcesDomainIds);
        body.withAccountAggregationSources(accountAggregationSourcesbody);
        body.withAggregatorType("ACCOUNT");
        body.withAggregatorName("example_aggregator");
        request.withBody(body);
        try {
            UpdateConfigurationAggregatorResponse response =
client.updateConfigurationAggregator(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

更新资源聚合器。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *
```

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = UpdateConfigurationAggregatorRequest()
        listDomainIdsAccountAggregationSources = [
            "3fd23817bd85480e8dc4f520075de348"
        ]
        accountAggregationSourcesbody = AccountAggregationSource(
            domain_ids=listDomainIdsAccountAggregationSources
        )
        request.body = ConfigurationAggregatorRequest(
            account_aggregation_sources=accountAggregationSourcesbody,
            aggregator_type="ACCOUNT",
            aggregator_name="example_aggregator"
        )
        response = client.update_configuration_aggregator(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

更新资源聚合器。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).
Build()

request := &model.UpdateConfigurationAggregatorRequest{}
var listDomainIdsAccountAggregationSources = []string{
    "3fd23817bd85480e8dc4f520075de348",
}
accountAggregationSourcesbody := &model.AccountAggregationSource{
    DomainIds: &listDomainIdsAccountAggregationSources,
}
request.Body = &model.ConfigurationAggregatorRequest{
    AccountAggregationSources: accountAggregationSourcesbody,
    AggregatorType: "ACCOUNT",
    AggregatorName: "example_aggregator",
}
response, err := client.UpdateConfigurationAggregator(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.6 删除资源聚合器

### 功能介绍

删除资源聚合器。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/resource-manager/domains/{domain\_id}/aggregators/{aggregator\_id}

表 4-469 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
aggregator_id	是	String	资源聚合器ID。 最大长度：32

## 请求参数

表 4-470 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

无

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeleteConfigurationAggregatorSolution {
```

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

    ICredential auth = new GlobalCredentials()
        .withAk(ak)
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    DeleteConfigurationAggregatorRequest request = new DeleteConfigurationAggregatorRequest();
    try {
        DeleteConfigurationAggregatorResponse response = client.deleteConfigurationAggregator(request);
        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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeleteConfigurationAggregatorRequest()
        response = client.delete_configuration_aggregator(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
204	操作成功。

## 错误码

请参见[错误码](#)。

### 4.8.7 创建资源聚合器授权

#### 功能介绍

给资源聚合器账号授予从源账号收集数据的权限。



## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v1/resource-manager/domains/{domain\_id}/aggregators/aggregation-authorization

表 4-471 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

## 请求参数

表 4-472 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-473 请求 Body 参数

参数	是否必选	参数类型	描述
authorized_account_id	是	String	要授权的资源聚合器的账号ID。 最小长度：1 最大长度：36

## 响应参数

状态码：200

表 4-474 响应 Body 参数

参数	参数类型	描述
aggregation_authorization_urn	String	资源聚合器授权标识符。

参数	参数类型	描述
authorized_account_id	String	授权的资源聚合器的账号ID。
created_at	String	资源聚合器授权的创建时间。

状态码： 400

表 4-475 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-476 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-477 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

给资源聚合器账号授予从源账号收集数据的权限。

```
PUT https://{endpoint}/v1/resource-manager/domains/{domain_id}/aggregators/aggregation-authorization
{
  "authorized_account_id": "3fd23817bd85480e8dc4f520075de348"
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "aggregation_authorization_urn" :
  "rms::e74e043fab784a45ad88f5ef6a4bcffc:aggregationAuthorization:fb51a91e798a4beb831c9be365aba4db",
  "authorized_account_id" : "3fd23817bd85480e8dc4f520075de348",
  "created_at" : "2020-01-18T02:49:12.465Z"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

给资源聚合器账号授予从源账号收集数据的权限。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class CreateAggregationAuthorizationSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

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

        CreateAggregationAuthorizationRequest request = new CreateAggregationAuthorizationRequest();
        AggregationAuthorizationRequest body = new AggregationAuthorizationRequest();
        body.withAuthorizedAccountId("3fd23817bd85480e8dc4f520075de348");
        request.withBody(body);
        try {
            CreateAggregationAuthorizationResponse response =
            client.createAggregationAuthorization(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

给资源聚合器账号授予从源账号收集数据的权限。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CreateAggregationAuthorizationRequest()
        request.body = AggregationAuthorizationRequest(
            authorized_account_id="3fd23817bd85480e8dc4f520075de348"
        )
        response = client.create_aggregation_authorization(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

给资源聚合器账号授予从源账号收集数据的权限。

```
package main

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

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

```
sk := os.Getenv("CLOUD_SDK_SK")

auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateAggregationAuthorizationRequest{}
request.Body = &model.AggregationAuthorizationRequest{
    AuthorizedAccountId: "3fd23817bd85480e8dc4f520075de348",
}
response, err := client.CreateAggregationAuthorization(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.8 查询资源聚合器授权列表

### 功能介绍

查询授权过的资源聚合器列表。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/aggregators/aggregation-authorization

表 4-478 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-479 Query 参数

参数	是否必选	参数类型	描述
account_id	否	String	授权的账号ID。 最大长度：36
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-480 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-481 响应 Body 参数

参数	参数类型	描述
aggregation_authorized	Array of <b>AggregationAuthorizationResp</b> objects	授权过的资源聚合器账号列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-482 AggregationAuthorizationResp

参数	参数类型	描述
aggregation_authorized_urn	String	资源聚合器授权标识符。
authorized_account_id	String	授权的资源聚合器的账号ID。
created_at	String	资源聚合器授权的创建时间。

表 4-483 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-484 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-485 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-486 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "aggregation_authorizations" : [ {
    "aggregation_authorization_urn" :
"rms::e74e043fab784a45ad88f5ef6a4bcffc:aggregationAuthorization:fb51a91e798a4beb831c9be365aba4db",
    "authorized_account_id" : "3fd23817bd85480e8dc4f520075de348",
    "created_at" : "2022-12-15T07:50:01.575Z"
  } ],
  "page_info" : {
    "current_count" : 1,
    "next_marker" : "CAESIDdjYTM0NDc4MzBiZDQ5YzliNjMwYTk5NjgwZWVmOTBmGgQ-0aw3"
  }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```



```
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListAggregationAuthorizationsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAggregationAuthorizationsRequest request = new ListAggregationAuthorizationsRequest();
        request.withAccountId("<account_id>");
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        try {
            ListAggregationAuthorizationsResponse response = client.listAggregationAuthorizations(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

```
try:
    request = ListAggregationAuthorizationsRequest()
    request.account_id = "<account_id>"
    request.limit = <limit>
    request.marker = "<marker>"
    response = client.list_aggregation_authorizations(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAggregationAuthorizationsRequest{}
    accountIdRequest := "<account_id>"
    request.AccountId = &accountIdRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListAggregationAuthorizations(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.9 删除资源聚合器授权

### 功能介绍

删除指定资源聚合器账号的授权。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/resource-manager/domains/{domain\_id}/aggregators/aggregation-authorization/{authorized\_account\_id}

表 4-487 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
authorized_account_id	是	String	授权的资源聚合器的账号ID。 最大长度：36

## 请求参数

表 4-488 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

无

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeleteAggregationAuthorizationSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        DeleteAggregationAuthorizationRequest request = new DeleteAggregationAuthorizationRequest();
        try {
            DeleteAggregationAuthorizationResponse response =
client.deleteAggregationAuthorization(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeleteAggregationAuthorizationRequest()
        response = client.delete_aggregation_authorization(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
204	操作成功。

## 错误码

请参见[错误码](#)。

## 4.8.10 查询所有挂起的聚合请求列表

### 功能介绍

查询所有挂起的聚合请求列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/aggregators/pending-aggregation-request

表 4-489 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-490 Query 参数

参数	是否必选	参数类型	描述
account_id	否	String	授权的账号ID。 最大长度：36
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-491 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-492 响应 Body 参数

参数	参数类型	描述
pending_aggregation_requests	Array of <b>PendingAggregationRequest</b> objects	挂起的聚合请求列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-493 PendingAggregationRequest

参数	参数类型	描述
requester_account_id	String	请求聚合数据的账号ID。

表 4-494 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-495 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403



表 4-496 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-497 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "pending_aggregation_requests" : [ {
    "requester_account_id" : "e74e043fab784a45ad88f5ef6a4bcffc"
  } ],
  "page_info" : {
    "current_count" : 1,
    "next_marker" : "CAESIDdjYTM0NDc4MzBiZDQ5YzliNjMwYTk5NjgwZWFMOTBmGgQ-0aw3"
  }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListPendingAggregationRequestsSolution {
```

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

    ICredential auth = new GlobalCredentials()
        .withAk(ak)
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    ListPendingAggregationRequestsRequest request = new ListPendingAggregationRequestsRequest();
    request.withAccountId("<account_id>");
    request.withLimit(<limit>);
    request.withMarker("<marker>");
    try {
        ListPendingAggregationRequestsResponse response =
client.listPendingAggregationRequests(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListPendingAggregationRequestsRequest()
        request.account_id = "<account_id>"
        request.limit = <limit>
```

```
request.marker = "<marker>"
response = client.list_pending_aggregation_requests(request)
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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPendingAggregationRequestsRequest{}
    accountIdRequest := "<account_id>"
    request.AccountId = &accountIdRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListPendingAggregationRequests(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.8.11 删除聚合器账号中挂起的授权请求

#### 功能介绍

删除聚合器账号中挂起的授权请求。

#### 调用方法

请参见[如何调用API](#)。

#### URI

DELETE /v1/resource-manager/domains/{domain\_id}/aggregators/pending-aggregation-request/{requester\_account\_id}

表 4-498 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
requester_account_id	是	String	请求聚合数据的账号ID。 最大长度：36

## 请求参数

表 4-499 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

无

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeletePendingAggregationRequestSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        DeletePendingAggregationRequestRequest request = new DeletePendingAggregationRequestRequest();
        try {
            DeletePendingAggregationRequestResponse response =
client.deletePendingAggregationRequest(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeletePendingAggregationRequestRequest()
        response = client.delete_pending_aggregation_request(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
204	操作成功。

## 错误码

请参见[错误码](#)。

## 4.8.12 查询聚合器中账号资源的计数

### 功能介绍

查询聚合器中账号资源的计数，支持通过过滤器和GroupByKey来统计资源数量。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/aggregate-discovered-resource-counts

表 4-500 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

## 请求参数

表 4-501 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-502 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID。 最小长度：1 最大长度：256
filter	否	ResourceCountsFilters object	资源计数过滤器。
group_by_key	是	String	用于对资源计数进行分组的键（RESOURCE_TYPE   ACCOUNT_ID）。 最小长度：1 最大长度：16

表 4-503 ResourceCountsFilters

参数	是否必选	参数类型	描述
account_id	否	String	账号ID。 最小长度：1 最大长度：36



参数	是否必选	参数类型	描述
resource_type	否	String	资源类型。 最小长度：1 最大长度：32
region_id	否	String	区域ID。 最小长度：1 最大长度：32

## 响应参数

状态码：200

表 4-504 响应 Body 参数

参数	参数类型	描述
group_by_key	String	资源计数进行分组的键。
grouped_resource_counts	Array of <b>GroupedResourceCount</b> objects	分组资源计数的列表。
total_discovered_resources	Integer	指定过滤器的资源聚合器中存在的资源总数。

表 4-505 GroupedResourceCount

参数	参数类型	描述
group_name	String	分组名称。
resource_count	Integer	资源数量。

状态码：400

表 4-506 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

**状态码： 403**

**表 4-507** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

**状态码： 500**

**表 4-508** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询聚合器中账号资源的计数。

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/aggregate-discovered-resource-counts
{
  "aggregator_id": "7ca3447830bd49c9b630a99680eaf90f",
  "group_by_key": "DOMAIN"
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "group_by_key": "DOMAIN",
  "grouped_resource_counts": [ {
    "group_name": "3fd23817bd85480e8dc4f520075de348",
    "resource_count": 23
  } ],
  "total_discovered_resources": 23
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询聚合器中账号资源的计数。

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

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowAggregateDiscoveredResourceCountsSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAggregateDiscoveredResourceCountsRequest request = new
        ShowAggregateDiscoveredResourceCountsRequest();
        AggregateDiscoveredResourceCountsRequest body = new
        AggregateDiscoveredResourceCountsRequest();
        body.withGroupByKey("DOMAIN");
        body.withAggregatorId("7ca3447830bd49c9b630a99680eaf90f");
        request.withBody(body);
        try {
            ShowAggregateDiscoveredResourceCountsResponse response =
            client.showAggregateDiscoveredResourceCounts(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

查询聚合器中账号资源的计数。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = ShowAggregateDiscoveredResourceCountsRequest()
    request.body = AggregateDiscoveredResourceCountsRequest(
        group_by_key="DOMAIN",
        aggregator_id="7ca3447830bd49c9b630a99680eaf90f"
    )
    response = client.show_aggregate_discovered_resource_counts(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

查询聚合器中账号资源的计数。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAggregateDiscoveredResourceCountsRequest{
        request.Body = &model.AggregateDiscoveredResourceCountsRequest{
            GroupByKey: "DOMAIN",
            AggregatorId: "7ca3447830bd49c9b630a99680eaf90f",
        }
    }
    response, err := client.ShowAggregateDiscoveredResourceCounts(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
```

```
    fmt.Println(err)
  }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.13 查询聚合器中资源的列表

### 功能介绍

查询资源聚合器中特定资源的列表。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/aggregate-discovered-resources

表 4-509 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-510 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-511 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-512 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID。 最小长度：1 最大长度：256
filter	否	ResourcesFilters object	资源计数过滤器。
provider	否	String	云服务类型。 最小长度：1 最大长度：20
resource_type	否	String	资源类型。 最小长度：1 最大长度：32

表 4-513 ResourcesFilters

参数	是否必选	参数类型	描述
account_id	否	String	账号ID。 最小长度：1 最大长度：36
region_id	否	String	区域ID。 最小长度：1 最大长度：36
resource_id	否	String	资源ID。 最小长度：1 最大长度：512
resource_name	否	String	资源名称。 最小长度：1 最大长度：256

## 响应参数

状态码：200

表 4-514 响应 Body 参数

参数	参数类型	描述
resource_identifiers	Array of <b>ResourceIdentifier</b> objects	资源信息列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-515 ResourceIdentifier

参数	参数类型	描述
resource_id	String	资源ID。 最小长度：1 最大长度：512
resource_name	String	资源名称。 最小长度：1 最大长度：256

参数	参数类型	描述
provider	String	云服务类型。 最小长度：1 最大长度：20
type	String	资源类型。 最小长度：1 最大长度：32
source_account_id	String	源账号ID。 最小长度：1 最大长度：36
region_id	String	资源所属区域。 最小长度：1 最大长度：36

表 4-516 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-517 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403



表 4-518 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-519 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询资源聚合器中特定资源类型的列表。

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/aggregate-discovered-resources

{
  "aggregator_id": "7ca3447830bd49c9b630a99680eaf90f",
  "provider": "vpc"
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "resource_identifiers": [ {
    "resource_id": "1a429d95-3723-4bfa-b54a-19f041f66fe5",
    "resource_name": "vpc1",
    "provider": "vpc",
    "type": "vpcs",
    "source_account_id": "3fd23817bd85480e8dc4f520075de348",
    "region_id": "region1"
  }, {
    "resource_id": "3d23d144-82e8-4c7c-b0f7-06e4f413d0ee",
    "resource_name": "vpc2",
    "provider": "vpc",
    "type": "vpcs",
    "source_account_id": "3fd23817bd85480e8dc4f520075de348",
    "region_id": "region2"
  } ],
  "page_info": {
    "current_count": 2,
    "next_marker": "CAESIDdjYTM0NDc4MzBiZDQ5YzliNjMwYTk5NjgwZWVmOTBmGgQ-0aw3"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询资源聚合器中特定资源类型的列表。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListAggregateDiscoveredResourcesSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAggregateDiscoveredResourcesRequest request = new ListAggregateDiscoveredResourcesRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        AggregateDiscoveredResourcesRequest body = new AggregateDiscoveredResourcesRequest();
        body.withProvider("vpc");
        body.withAggregatorId("7ca3447830bd49c9b630a99680eaf90f");
        request.withBody(body);
        try {
            ListAggregateDiscoveredResourcesResponse response =
client.listAggregateDiscoveredResources(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

查询资源聚合器中特定资源类型的列表。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListAggregateDiscoveredResourcesRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.body = AggregateDiscoveredResourcesRequest(
            provider="vpc",
            aggregator_id="7ca3447830bd49c9b630a99680eaf90f"
        )
        response = client.list_aggregate_discovered_resources(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

查询资源聚合器中特定资源类型的列表。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
```

```
WithRegion(region.ValueOf("<YOUR REGION>")).
WithCredential(auth).
Build()

request := &model.ListAggregateDiscoveredResourcesRequest{}
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
providerAggregateDiscoveredResourcesRequest:= "vpc"
request.Body = &model.AggregateDiscoveredResourcesRequest{
    Provider: &providerAggregateDiscoveredResourcesRequest,
    AggregatorId: "7ca3447830bd49c9b630a99680eaf90f",
}
response, err := client.ListAggregateDiscoveredResources(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.8.14 查询源账号中资源的详情

#### 功能介绍

查询源账号中特定资源的详情。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-resource-config

表 4-520 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

## 请求参数

表 4-521 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-522 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID。 最小长度：1 最大长度：256
resource_identifier	是	<b>ResourceIdentifier</b> object	资源详细信息的列表。

表 4-523 ResourceIdentifier

参数	是否必选	参数类型	描述
resource_id	是	String	资源ID。 最小长度：1 最大长度：512
resource_name	否	String	资源名称。 最小长度：1 最大长度：256
provider	是	String	云服务类型。 最小长度：1 最大长度：20

参数	是否必选	参数类型	描述
type	是	String	资源类型。 最小长度：1 最大长度：32
source_account_id	是	String	源账号ID。 最小长度：1 最大长度：36
region_id	是	String	资源所属区域。 最小长度：1 最大长度：36

## 响应参数

状态码：200

表 4-524 响应 Body 参数

参数	参数类型	描述
resource_id	String	资源ID。
aggregator_id	String	聚合器ID。
aggregator_domain_id	String	聚合器账号。
domain_id	String	聚合资源所属账号的ID。
ep_id	String	企业项目ID。
provider	String	云服务名称。
type	String	资源类型。
name	String	资源名称。
region_id	String	区域ID。
project_id	String	Openstack中的项目ID。
created	String	资源创建时间。
updated	String	资源更新时间。
tags	Map<String,String>	资源标签。
properties	Map<String,Object>	资源详细属性。

状态码： 400

表 4-525 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-526 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-527 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询源账号中特定资源的详情。

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/aggregators/aggregate-resource-config
{
  "aggregator_id" : "7ca3447830bd49c9b630a99680eaf90f",
  "resource_identifier" : {
    "resource_id" : "1a429d95-3723-4bfa-b54a-19f041f66fe5",
    "provider" : "vpc",
    "type" : "vpcs",
    "source_account_id" : "3fd23817bd85480e8dc4f520075de348",
    "region_id" : "regionid1"
  }
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "resource_id": "1a429d95-3723-4bfa-b54a-19f041f66fe5",
  "aggregator_id": "7ca3447830bd49c9b630a99680eaf90f",
  "aggregator_domain_id": "e74e043fab784a45ad88f5ef6a4bcffc",
  "domain_id": "3fd23817bd85480e8dc4f520075de348",
  "ep_id": 0,
  "provider": "vpc",
  "type": "vpcs",
  "name": "vpc-xxx",
  "region_id": "regionid1",
  "project_id": "377cd852c57e4e949fc27c4882310364",
  "created": "2022-10-24T07:03:40.000+00:00",
  "updated": "2022-10-24T07:04:17.000+00:00",
  "properties": {
    "cidr": "192.168.0.0/16",
    "status": "OK"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询源账号中特定资源的详情。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowAggregateResourceConfigSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAggregateResourceConfigRequest request = new ShowAggregateResourceConfigRequest();
        AggregateResourceConfigRequest body = new AggregateResourceConfigRequest();
        ResourceIdentifier resourceIdentifierbody = new ResourceIdentifier();
        resourceIdentifierbody.withResourceId("1a429d95-3723-4bfa-b54a-19f041f66fe5")
            .withProvider("vpc")
            .withType("vpcs")
            .withSourceAccountId("3fd23817bd85480e8dc4f520075de348")
            .withRegionId("regionid1");
        body.withResourceIdentifier(resourceIdentifierbody);
    }
}
```



```
body.withAggregatorId("7ca3447830bd49c9b630a99680eaf90f");
request.withBody(body);
try {
    ShowAggregateResourceConfigResponse response = client.showAggregateResourceConfig(request);
    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

查询源账号中特定资源的详情。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowAggregateResourceConfigRequest()
        resourceIdentifierbody = ResourceIdentifier(
            resource_id="1a429d95-3723-4bfa-b54a-19f041f66fe5",
            provider="vpc",
            type="vpcs",
            source_account_id="3fd23817bd85480e8dc4f520075de348",
            region_id="regionid1"
        )
        request.body = AggregateResourceConfigRequest(
            resource_identifier=resourceIdentifierbody,
            aggregator_id="7ca3447830bd49c9b630a99680eaf90f"
        )
        response = client.show_aggregate_resource_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

查询源账号中特定资源的详情。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAggregateResourceConfigRequest{
        resourceIdentifierbody := &model.ResourceIdentifier{
            ResourceId: "1a429d95-3723-4bfa-b54a-19f041f66fe5",
            Provider: "vpc",
            Type: "vpcs",
            SourceAccountId: "3fd23817bd85480e8dc4f520075de348",
            RegionId: "regionid1",
        }
    }
    request.Body = &model.AggregateResourceConfigRequest{
        ResourceIdentifier: resourceIdentifierbody,
        AggregatorId: "7ca3447830bd49c9b630a99680eaf90f",
    }
    response, err := client.ShowAggregateResourceConfig(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。

状态码	描述
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.15 对指定聚合器执行高级查询

### 功能介绍

对指定聚合器执行高级查询。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/domains/{domain\_id}/aggregators/{aggregator\_id}/run-query

表 4-528 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
aggregator_id	是	String	资源聚合器ID。 最大长度：32

### 请求参数

表 4-529 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-530 请求 Body 参数

参数	是否必选	参数类型	描述
expression	是	String	ResourceQL表达式。 最小长度：1 最大长度：4096

## 响应参数

状态码：200

表 4-531 响应 Body 参数

参数	参数类型	描述
query_info	<b>QueryInfo</b> object	ResourceQL查询字段。
results	Array of objects	ResourceQL查询结果。

表 4-532 QueryInfo

参数	参数类型	描述
select_fields	Array of strings	ResourceQL查询字段。

状态码：400

表 4-533 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-534 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	错误消息内容。

状态码： 500

表 4-535 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询当前组织下所有虚拟机的ID。

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/aggregators/{aggregator_id}/run-query
{
  "expression" : "select id from aggregator_resources where provider = 'ecs' and type = 'cloudservers'"
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "query_info" : {
    "select_fields" : [ "id" ]
  },
  "results" : [ {
    "id" : "91252cc9-bfd9-0709-0912-56b397e0ba3f"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询当前组织下所有虚拟机的ID。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
```

```
import com.huaweicloud.sdk.config.v1.model.*;

public class RunAggregateResourceQuerySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        RunAggregateResourceQueryRequest request = new RunAggregateResourceQueryRequest();
        QueryRunRequestBody body = new QueryRunRequestBody();
        body.withExpression("select id from aggregator_resources where provider = 'ecs' and type =
'cloudservers'");
        request.withBody(body);
        try {
            RunAggregateResourceQueryResponse response = client.runAggregateResourceQuery(request);
            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

查询当前组织下所有虚拟机的ID。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

```
.build()

try:
    request = RunAggregateResourceQueryRequest()
    request.body = QueryRunRequestBody(
        expression="select id from aggregator_resources where provider = 'ecs' and type = 'cloudservers'"
    )
    response = client.run_aggregate_resource_query(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

查询当前组织下所有虚拟机的ID。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RunAggregateResourceQueryRequest{}
    request.Body = &model.QueryRunRequestBody{
        Expression: "select id from aggregator_resources where provider = 'ecs' and type = 'cloudservers'",
    }
    response, err := client.RunAggregateResourceQuery(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.8.16 查询聚合器中一个或多个账号的合规概况

#### 功能介绍

查询聚合器中一个或多个账号的合规和不合规规则数。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/  
policy-states/compliance-summary

表 4-536 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

#### 请求参数

表 4-537 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。



表 4-538 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID 最小长度：1 最大长度：256
account_id	否	String	源账号ID 最小长度：1 最大长度：36
group_by_key	否	String	用于对资源计数进行分组的键（DOMAIN）。 最小长度：1 最大长度：16

## 响应参数

状态码：200

表 4-539 响应 Body 参数

参数	参数类型	描述
results	Array of <a href="#">AggregatePolicyComplianceSummaryResult</a> objects	合规总结结果列表

表 4-540 AggregatePolicyComplianceSummaryResult

参数	参数类型	描述
resource_details	<a href="#">PolicyComplianceSummaryUnit</a> object	规则总结详情。
assignment_details	<a href="#">PolicyComplianceSummaryUnit</a> object	规则总结详情。
group_name	String	分组名称
group_account_name	String	账号名称

表 4-541 PolicyComplianceSummaryUnit

参数	参数类型	描述
compliant_count	Integer	合规数量。
non_compliant_count	Integer	不合规数量。

状态码： 400

表 4-542 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-543 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-544 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询聚合器中一个或多个账号的合规概况

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/policy-states/compliance-summary
```

```
{  
  "aggregator_id" : "7ca3447830bd49c9b630a99680eaf90f",
```

```
"group_by_key" : "DOMAIN"  
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{  
  "results" : [ {  
    "resource_details" : {  
      "compliant_count" : null,  
      "non_compliant_count" : 70  
    },  
    "assignment_details" : {  
      "compliant_count" : 0,  
      "non_compliant_count" : 1  
    },  
    "group_name" : "39f4a9434e0c4bf3806e5aff54adea39",  
    "group_account_name" : "demo1"  
  }, {  
    "resource_details" : {  
      "compliant_count" : null,  
      "non_compliant_count" : 0  
    },  
    "assignment_details" : {  
      "compliant_count" : 1,  
      "non_compliant_count" : 0  
    },  
    "group_name" : "789d4ac10eda4d9b8e090f5c2b1e4459",  
    "group_account_name" : "demo2"  
  } ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询聚合器中一个或多个账号的合规概况

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class ShowAggregatePolicyStateComplianceSummarySolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
    }  
}
```

```
ICredential auth = new GlobalCredentials()
    .withAk(ak)
    .withSk(sk);

ConfigClient client = ConfigClient.newBuilder()
    .withCredential(auth)
    .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
    .build();
ShowAggregatePolicyStateComplianceSummaryRequest request = new
ShowAggregatePolicyStateComplianceSummaryRequest();
AggregatePolicyStatesRequest body = new AggregatePolicyStatesRequest();
body.withGroupByKey("DOMAIN");
body.withAggregatorId("7ca3447830bd49c9b630a99680eaf90f");
request.withBody(body);
try {
    ShowAggregatePolicyStateComplianceSummaryResponse response =
client.showAggregatePolicyStateComplianceSummary(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

### 查询聚合器中一个或多个账号的合规概况

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowAggregatePolicyStateComplianceSummaryRequest()
        request.body = AggregatePolicyStatesRequest(
            group_by_key="DOMAIN",
            aggregator_id="7ca3447830bd49c9b630a99680eaf90f"
        )
        response = client.show_aggregate_policy_state_compliance_summary(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

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

## Go

### 查询聚合器中一个或多个账号的合规概况

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAggregatePolicyStateComplianceSummaryRequest{}
    groupByKeyAggregatePolicyStatesRequest := "DOMAIN"
    request.Body = &model.AggregatePolicyStatesRequest{
        GroupByKey: &groupByKeyAggregatePolicyStatesRequest,
        AggregatorId: "7ca3447830bd49c9b630a99680eaf90f",
    }
    response, err := client.ShowAggregatePolicyStateComplianceSummary(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。

状态码	描述
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.17 查询聚合合规规则列表

### 功能介绍

查询合规和不合规规则的列表，其中包含合规和不合规规则的资源数量。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/  
policy-assignments/compliance

表 4-545 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-546 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-547 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-548 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID 最小长度：1 最大长度：256
filter	否	AggregatePolicyAssignmentsFilters object	聚合合规规则过滤器。

表 4-549 AggregatePolicyAssignmentsFilters

参数	是否必选	参数类型	描述
account_id	否	String	源账号ID。 最小长度：1 最大长度：36
compliance_state	否	String	合规结果。
policy_assignment_name	否	String	合规规则名称 最大长度：64

## 响应参数

状态码：200

表 4-550 响应 Body 参数

参数	参数类型	描述
aggregate_policy_assignments	Array of <b>AggregatePolicyAssignments</b> objects	聚合合规规则的列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-551 AggregatePolicyAssignments

参数	参数类型	描述
policy_assignment_id	String	合规规则ID
policy_assignment_name	String	合规规则名称
compliance	<b>Compliance</b> object	合规规则合规性
account_id	String	源账号ID。
account_name	String	账号名称。

表 4-552 Compliance

参数	参数类型	描述
compliance_state	String	合规结果。
resource_details	<b>PolicyComplianceSummaryUnit</b> object	规则总结详情。

表 4-553 PolicyComplianceSummaryUnit

参数	参数类型	描述
compliant_count	Integer	合规数量。
non_compliant_count	Integer	不合规数量。



表 4-554 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-555 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-556 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-557 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询聚合合规规则的列表

POST https://{endpoint}/v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/policy-assignments/compliance

```
{  
  "aggregator_id": "7ca3447830bd49c9b630a99680eaf90f"  
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{  
  "aggregate_policy_assignments": [ {  
    "policy_assignment_id": "62bdbe5913c6836025b1e3af",  
    "policy_assignment_name": "volumes-encrypted-check",  
    "compliance": {  
      "compliance_state": "Compliant",  
      "resource_details": {  
        "compliant_count": 18,  
        "non_compliant_count": 0  
      }  
    }  
  },  
  "account_id": "789d4ac10eda4d9b8e090f5c2b1e4459",  
  "account_name": "demo1"  
}, {  
  "policy_assignment_id": "631ea13f57e0427679a1794d",  
  "policy_assignment_name": "cts-lts-enable",  
  "compliance": {  
    "compliance_state": "NonCompliant",  
    "resource_details": {  
      "compliant_count": 28,  
      "non_compliant_count": 2  
    }  
  },  
  "account_id": "83240c3f2ec1465aaa0124c87810ce65",  
  "account_name": "demo2"  
}],  
  "page_info": {  
    "current_count": 2,  
    "next_marker": "CAESIDdjYTM0NDc4MzBiZDQ5YzliNjMwYTk5NjgwZWZmOTBmGgQ-0aw3"  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询聚合合规规则的列表

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class ListAggregateComplianceByPolicyAssignmentSolution {  
  public static void main(String[] args) {
```

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

ICredential auth = new GlobalCredentials()
    .withAk(ak)
    .withSk(sk);

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

ListAggregateComplianceByPolicyAssignmentRequest request = new
ListAggregateComplianceByPolicyAssignmentRequest();
request.withLimit(<limit>);
request.withMarker("<marker>");
AggregatePolicyAssignmentsRequest body = new AggregatePolicyAssignmentsRequest();
body.withAggregatorId("7ca3447830bd49c9b630a99680eaf90f");
request.withBody(body);
try {
    ListAggregateComplianceByPolicyAssignmentResponse response =
client.listAggregateComplianceByPolicyAssignment(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

### 查询聚合合规规则的列表

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
```

```
request = ListAggregateComplianceByPolicyAssignmentRequest()
request.limit = <limit>
request.marker = "<marker>"
request.body = AggregatePolicyAssignmentsRequest(
    aggregator_id="7ca3447830bd49c9b630a99680eaf90f"
)
response = client.list_aggregate_compliance_by_policy_assignment(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 查询聚合合规规则的列表

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAggregateComplianceByPolicyAssignmentRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    request.Body = &model.AggregatePolicyAssignmentsRequest{
        AggregatorId: "7ca3447830bd49c9b630a99680eaf90f",
    }
    response, err := client.ListAggregateComplianceByPolicyAssignment(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

### 4.8.18 查询指定聚合合规规则的资源评估结果详情

#### 功能介绍

查询指定聚合合规规则的资源评估结果详情。包含评估的资源，以及每个资源是否符合规则。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/policy-states/compliance-details

表 4-558 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-559 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200

参数	是否必选	参数类型	描述
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

表 4-560 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-561 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID 最小长度：1 最大长度：256
account_id	否	String	源账号ID 最小长度：1 最大长度：36
compliance_state	否	String	合规结果。
policy_assignment_name	否	String	合规规则名称 最大长度：64
resource_name	否	String	资源名称 最小长度：1 最大长度：256
resource_id	否	String	资源ID 最小长度：1 最大长度：512

## 响应参数

状态码： 200

表 4-562 响应 Body 参数

参数	参数类型	描述
policy_states	Array of <b>PolicyState</b> objects	合规结果查询返回值
page_info	<b>PageInfo</b> object	分页对象。

表 4-563 PolicyState

参数	参数类型	描述
domain_id	String	用户ID。
region_id	String	资源区域ID。
resource_id	String	资源ID。
resource_name	String	资源名称。
resource_provider	String	云服务名称。
resource_type	String	资源类型。
trigger_type	String	触发器类型，可选值为"resource"和"period"。
compliance_state	String	合规结果。
policy_assignment_id	String	规则ID。
policy_assignment_name	String	规则名称。
policy_definition_id	String	策略ID。
evaluation_time	String	合规状态评估时间。

表 4-564 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-565 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-566 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-567 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询指定聚合合规规则的资源评估结果详情

POST https://{endpoint}/v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/policy-states/compliance-details



```
{
  "aggregator_id": "7ca3447830bd49c9b630a99680eaf90f",
  "account_id": "39f4a9434e0c4bf3806e5aff54adea39",
  "policy_assignment_name": "required-tag-check"
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "policy_states": [ {
    "domain_id": "39f4a9434e0c4bf3806e5aff54adea39",
    "region_id": "region1",
    "resource_id": "04d5a1a31c314ebba8c73daa66c6cd12",
    "resource_name": "group1",
    "resource_provider": "iam",
    "resource_type": "groups",
    "trigger_type": "resource",
    "compliance_state": "NonCompliant",
    "policy_assignment_id": "64104fff86952b3af614cabb",
    "policy_assignment_name": "required-tag-check",
    "policy_definition_id": "5fa9f89b6eed194ccb2c04db",
    "evaluation_time": 1678790655196
  }, {
    "domain_id": "39f4a9434e0c4bf3806e5aff54adea39",
    "region_id": "global",
    "resource_id": "06c113bd3edf40eabc552d29a775b9cf",
    "resource_name": "group2",
    "resource_provider": "iam",
    "resource_type": "groups",
    "trigger_type": "resource",
    "compliance_state": "NonCompliant",
    "policy_assignment_id": "64104fff86952b3af614cabb",
    "policy_assignment_name": "required-tag-check",
    "policy_definition_id": "5fa9f89b6eed194ccb2c04db",
    "evaluation_time": 1678790655197
  } ],
  "page_info": {
    "current_count": 2,
    "next_marker": "CAESIDdjYTM0NDc4MzBiZDQ5YzliNjMwYTk5NjgwZWZmOTBmGgQ-0aw3"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询指定聚合合规规则的资源评估结果详情

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;
```

```
public class ShowAggregateComplianceDetailsByPolicyAssignmentSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

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

        ShowAggregateComplianceDetailsByPolicyAssignmentRequest request = new
        ShowAggregateComplianceDetailsByPolicyAssignmentRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        AggregateComplianceDetailRequest body = new AggregateComplianceDetailRequest();
        body.withPolicyAssignmentName("required-tag-check");
        body.withAccountId("39f4a9434e0c4bf3806e5aff54adea39");
        body.withAggregatorId("7ca3447830bd49c9b630a99680eaf90f");
        request.withBody(body);
        try {
            ShowAggregateComplianceDetailsByPolicyAssignmentResponse response =
            client.showAggregateComplianceDetailsByPolicyAssignment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

### 查询指定聚合合规规则的资源评估结果详情

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \
```

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

try:
    request = ShowAggregateComplianceDetailsByPolicyAssignmentRequest()
    request.limit = <limit>
    request.marker = "<marker>"
    request.body = AggregateComplianceDetailRequest(
        policy_assignment_name="required-tag-check",
        account_id="39f4a9434e0c4bf3806e5aff54adea39",
        aggregator_id="7ca3447830bd49c9b630a99680eaf90f"
    )
    response = client.show_aggregate_compliance_details_by_policy_assignment(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 查询指定聚合合规规则的资源评估结果详情

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAggregateComplianceDetailsByPolicyAssignmentRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    policyAssignmentNameAggregateComplianceDetailRequest := "required-tag-check"
    accountIdAggregateComplianceDetailRequest := "39f4a9434e0c4bf3806e5aff54adea39"
    request.Body = &model.AggregateComplianceDetailRequest{
        PolicyAssignmentName: &policyAssignmentNameAggregateComplianceDetailRequest,
        AccountId: &accountIdAggregateComplianceDetailRequest,
        AggregatorId: "7ca3447830bd49c9b630a99680eaf90f",
    }
    response, err := client.ShowAggregateComplianceDetailsByPolicyAssignment(request)
```

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.8.19 查询指定聚合合规规则详情

### 功能介绍

查询指定聚合合规规则的详情。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/policy-assignment/detail

表 4-568 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度： <b>36</b>

## 请求参数

表 4-569 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-570 请求 Body 参数

参数	是否必选	参数类型	描述
aggregator_id	是	String	资源聚合器ID 最小长度：1 最大长度：256
account_id	是	String	源账号ID 最小长度：1 最大长度：36
policy_assignment_id	是	String	合规规则ID 最大长度：36

## 响应参数

状态码：200

表 4-571 响应 Body 参数

参数	参数类型	描述
policy_assignment_type	String	规则类型，包括预定义合规规则(builtin)和用户自定义合规规则(custom)
id	String	规则ID。
name	String	规则名称。
description	String	规则描述。
policy_filter	<b>PolicyFilterDefinition</b> object	规则过滤器。
period	String	周期值，可选值：One_Hour, Three_Hours, Six_Hours, Twelve_Hours, TwentyFour_Hours

参数	参数类型	描述
state	String	规则状态。
created	String	规则创建时间。
updated	String	规则更新时间。
policy_definition_id	String	规则的策略ID。
custom_policy	<b>CustomPolicy</b> object	自定义合规规则
parameters	Map<String, <b>PolicyParameter</b> >	规则参数。
created_by	String	规则的创建者

表 4-572 PolicyFilterDefinition

参数	参数类型	描述
region_id	String	区域ID。 最大长度：128
resource_provider	String	云服务名称。 最大长度：128
resource_type	String	资源类型。 最大长度：128
resource_id	String	资源ID。 最大长度：512
tag_key	String	标签键。 最大长度：128
tag_value	String	标签值。 最大长度：256

表 4-573 CustomPolicy

参数	参数类型	描述
function_urn	String	自定义函数的URN 最大长度：1024
auth_type	String	自定义合规规则调用function方式

参数	参数类型	描述
auth_value	Map<String, Object>	自定义合规规则调用function方式的参数值

表 4-574 PolicyParameterValue

参数	参数类型	描述
value	Object	规则参数值。

状态码： 400

表 4-575 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-576 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-577 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

查询指定聚合合规规则详情

POST https://{endpoint}/v1/resource-manager/domains/{domain\_id}/aggregators/aggregate-data/policy-assignment/detail

```
{
  "aggregator_id" : "7ca3447830bd49c9b630a99680eaf90f",
  "account_id" : "39f4a9434e0c4bf3806e5aff54adea39",
  "policy_assignment_id" : "64104fff86952b3af614cabb"
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "policy_assignment_type" : "builtin",
  "id" : "64104fff86952b3af614cabb",
  "name" : "required-tag-check",
  "description" : "指定一个标签，不具有此标签的资源，视为“不合规”",
  "policy_filter" : {
    "region_id" : null,
    "resource_provider" : null,
    "resource_type" : null,
    "resource_id" : null,
    "tag_key" : null,
    "tag_value" : null
  },
  "period" : null,
  "state" : "Enabled",
  "created" : "2023-04-03T06:40:12.846Z",
  "updated" : "2023-04-03T06:40:12.846Z",
  "policy_definition_id" : "5fa9f89b6eed194ccb2c04db",
  "created_by" : "customer"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

查询指定聚合合规规则详情

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowAggregatePolicyAssignmentDetailSolution {

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

        ICredential auth = new GlobalCredentials()
```



```
        .withAk(ak)
        .withSk(sk);

    ConfigClient client = ConfigClient.newBuilder()
        .withCredential(auth)
        .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowAggregatePolicyAssignmentDetailRequest request = new
ShowAggregatePolicyAssignmentDetailRequest();
    AggregatePolicyAssignmentDetailRequest body = new AggregatePolicyAssignmentDetailRequest();
    body.withPolicyAssignmentId("64104fff86952b3af614cabb");
    body.withAccountId("39f4a9434e0c4bf3806e5aff54adea39");
    body.withAggregatorId("7ca3447830bd49c9b630a99680eaf90f");
    request.withBody(body);
    try {
        ShowAggregatePolicyAssignmentDetailResponse response =
client.showAggregatePolicyAssignmentDetail(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } 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

### 查询指定聚合合规规则详情

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowAggregatePolicyAssignmentDetailRequest()
        request.body = AggregatePolicyAssignmentDetailRequest(
            policy_assignment_id="64104fff86952b3af614cabb",
            account_id="39f4a9434e0c4bf3806e5aff54adea39",
            aggregator_id="7ca3447830bd49c9b630a99680eaf90f"
        )
        response = client.show_aggregate_policy_assignment_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

### 查询指定聚合合规规则详情

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAggregatePolicyAssignmentDetailRequest{}
    request.Body = &model.AggregatePolicyAssignmentDetailRequest{
        PolicyAssignmentId: "64104fff86952b3af614cabb",
        AccountId: "39f4a9434e0c4bf3806e5aff54adea39",
        AggregatorId: "7ca3447830bd49c9b630a99680eaf90f",
    }
    response, err := client.ShowAggregatePolicyAssignmentDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。

状态码	描述
400	输入参数不合法。
403	用户认证失败或无操作权限。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9 合规规则包

### 4.9.1 列举合规规则包

#### 功能介绍

列举用户的合规规则包。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v1/resource-manager/domains/{domain\_id}/conformance-packs

表 4-578 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-579 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200

参数	是否必选	参数类型	描述
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
conformance_pack_name	否	String	合规规则包名称。 最小长度：1 最大长度：64

## 请求参数

表 4-580 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-581 响应 Body 参数

参数	参数类型	描述
value	Array of <a href="#">Conformance Pack</a> objects	合规规则包查询列表。
page_info	<a href="#">PageInfo</a> object	分页对象。

表 4-582 ConformancePack

参数	参数类型	描述
id	String	合规规则包ID。
name	String	合规规则包名称。

参数	参数类型	描述
stack_id	String	资源栈 ( stack ) 的唯一ID。 最小长度: 1 最大长度: 64
stack_name	String	资源栈 ( stack ) 的名称。 最小长度: 1 最大长度: 64
deployment_id	String	部署ID。
created_at	String	合规规则包创建时间。
updated_at	String	合规规则包更新时间。
status	String	合规规则包部署状态。
error_message	String	部署或删除合规规则包错误时的错误信息
vars_structure	Array of <b>VarsStructure</b> objects	合规规则包参数。
created_by	String	创建者

表 4-583 VarsStructure

参数	参数类型	描述
var_key	String	参数名称。 最小长度: 1 最大长度: 64
var_value	Object	参数的值。

表 4-584 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值: 0 最大值: 200
next_marker	String	下一页的marker。 最小长度: 4 最大长度: 400

状态码： 400

表 4-585 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-586 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-587 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [ {
    "id": "a0123456789",
    "name": "conformance_pack_name-e35r",
    "stack_id": "stack-id-1234",
    "stack_name": "stack-name-1234",
    "deployment_id": "deployment_id_1234",
    "created_at": "2023-02-28T09:39:19Z",
    "updated_at": "2023-02-28T09:39:19Z",
    "status": "CREATE_FAILED",
    "error_message": "policy assignments exceed quota",
    "vars_structure": [ {
      "var_key": "test",
```

```
    "var_value" : "test"
  } ]
} ],
"page_info" : {
  "current_count" : 1,
  "next_marker" : 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListConformancePacksSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListConformancePacksRequest request = new ListConformancePacksRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withConformancePacksName("<conformance_pack_name>");
        try {
            ListConformancePacksResponse response = client.listConformancePacks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListConformancePacksRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.conformance_pack_name = "<conformance_pack_name>"
        response = client.list_conformance_packs(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```



```
request := &model.ListConformancePacksRequest{}
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
conformancePacksNameRequest:= "<conformance_pack_name>"
request.ConformancePacksName = &conformancePacksNameRequest
response, err := client.ListConformancePacks(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.2 创建合规规则包

### 功能介绍

创建新的合规规则包。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/domains/{domain\_id}/conformance-packs

表 4-588 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

## 请求参数

表 4-589 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

表 4-590 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	合规规则包名称。 最小长度：0 最大长度：64
agency_name	否	String	委托名称，该委托需要授权云服务RFS调用合规规则的创建、更新、删除接口。 最小长度：0 最大长度：64
template_key	否	String	预定义合规包模板名称。 最大长度：128
template_body	否	String	自定义合规包内容。 最小长度：1 最大长度：51200
template_uri	否	String	合规包模板OBS地址。 最小长度：0 最大长度：1024
vars_structure	否	Array of <b>VarsStructure</b> objects	合规规则包参数。 数组长度：0 - 50

表 4-591 VarsStructure

参数	是否必选	参数类型	描述
var_key	否	String	参数名称。 最小长度：1 最大长度：64
var_value	否	Object	参数的值。

## 响应参数

状态码：200

表 4-592 响应 Body 参数

参数	参数类型	描述
id	String	合规规则包ID。
name	String	合规规则包名称。
stack_id	String	资源栈（stack）的唯一ID。 最小长度：1 最大长度：64
stack_name	String	资源栈（stack）的名称。 最小长度：1 最大长度：64
deployment_id	String	部署ID。
created_at	String	合规规则包创建时间。
updated_at	String	合规规则包更新时间。
status	String	合规规则包部署状态。
error_message	String	部署或删除合规规则包错误时的错误信息
vars_structure	Array of <b>VarsStructure</b> objects	合规规则包参数。
created_by	String	创建者

表 4-593 VarsStructure

参数	参数类型	描述
var_key	String	参数名称。 最小长度：1 最大长度：64
var_value	Object	参数的值。

状态码：400

表 4-594 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-595 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-596 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

根据预定义模板创建一个合规规则包。

```
POST https://{endpoint}/v1/resource-manager/domains/{domain_id}/conformance-packs
{
  "name": "conformance_pack_name",
```

```
"agency_name" : "config_conformance_pack_template",
"template_key" : "example-001.yaml",
"template_body" : null,
"template_uri" : null,
"vars_structure" : [ {
  "var_key" : "test",
  "var_value" : "test"
} ]
}
```

## 响应示例

**状态码： 200**

操作成功。

```
{
  "id" : "a0123456789",
  "name" : "conformance_pack_name-e35r",
  "stack_id" : "stack-id-1234",
  "stack_name" : "stack-name-1234",
  "deployment_id" : "deployment_id_1234",
  "created_at" : "2023-02-28T09:39:19Z",
  "updated_at" : "2023-02-28T09:39:19Z",
  "status" : "CREATE_FAILED",
  "error_message" : "policy assignments exceed quota",
  "vars_structure" : [ {
    "var_key" : "test",
    "var_value" : "test"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

根据预定义模板创建一个合规规则包。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

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

public class CreateConformancePackSolution {

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

        ICredential auth = new GlobalCredentials()
```

```
.withAk(ak)
.withSk(sk);

ConfigClient client = ConfigClient.newBuilder()
    .withCredential(auth)
    .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
    .build();
CreateConformancePackRequest request = new CreateConformancePackRequest();
ConformancePackRequestBody body = new ConformancePackRequestBody();
List<VarsStructure> listbodyVarsStructure = new ArrayList<>();
listbodyVarsStructure.add(
    new VarsStructure()
        .withVarKey("test")
        .withVarValue("test")
);
body.withVarsStructure(listbodyVarsStructure);
body.withTemplateKey("example-001.yaml");
body.withAgencyName("config_conformance_pack_template");
body.withName("conformance_pack_name");
request.withBody(body);
try {
    CreateConformancePackResponse response = client.createConformancePack(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

根据预定义模板创建一个合规规则包。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CreateConformancePackRequest()
        listVarsStructurebody = [
            VarsStructure(
                var_key="test",
```

```
        var_value="test"
    )
]
request.body = ConformancePackRequestBody(
    vars_structure=listVarsStructurebody,
    template_key="example-001.yaml",
    agency_name="config_conformance_pack_template",
    name="conformance_pack_name"
)
response = client.create_conformance_pack(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

根据预定义模板创建一个合规规则包。

```
package main

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

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConformancePackRequest{
        varKeyVarsStructure:= "test"
        varValueVarsStructure:= "test"
        var varValueVarsStructureInterface interface{} = varValueVarsStructure
        var listVarsStructurebody = []model.VarsStructure{
            {
                VarKey: &varKeyVarsStructure,
                VarValue: &varValueVarsStructureInterface,
            },
        }
        templateKeyConformancePackRequestBody:= "example-001.yaml"
        agencyNameConformancePackRequestBody:= "config_conformance_pack_template"
        request.Body = &model.ConformancePackRequestBody{
            VarsStructure: &listVarsStructurebody,
            TemplateKey: &templateKeyConformancePackRequestBody,
            AgencyName: &agencyNameConformancePackRequestBody,
            Name: "conformance_pack_name",
        }
    }
```

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.3 查看合规规则包

### 功能介绍

根据ID获取单个合规规则包。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/conformance-packs/{conformance\_pack\_id}

表 4-597 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
conformance_pack_id	是	String	合规规则包ID。 最大长度：36



## 请求参数

表 4-598 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码： 200

表 4-599 响应 Body 参数

参数	参数类型	描述
id	String	合规规则包ID。
name	String	合规规则包名称。
stack_id	String	资源栈（stack）的唯一ID。 最小长度：1 最大长度：64
stack_name	String	资源栈（stack）的名称。 最小长度：1 最大长度：64
deployment_id	String	部署ID。
created_at	String	合规规则包创建时间。
updated_at	String	合规规则包更新时间。
status	String	合规规则包部署状态。
error_message	String	部署或删除合规规则包错误时的错误信息
vars_structure	Array of <b>VarsStructure</b> objects	合规规则包参数。
created_by	String	创建者

表 4-600 VarsStructure

参数	参数类型	描述
var_key	String	参数名称。 最小长度：1 最大长度：64
var_value	Object	参数的值。

状态码：400

表 4-601 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-602 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-603 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-604 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "id": "a0123456789",
  "name": "conformance_pack_name-e35r",
  "stack_id": "stack-id-1234",
  "stack_name": "stack-name-1234",
  "deployment_id": "deployment_id_1234",
  "created_at": "2023-02-28T09:39:19Z",
  "updated_at": "2023-02-28T09:39:19Z",
  "status": "CREATE_FAILED",
  "error_message": "policy assignments exceed quota",
  "vars_structure": [ {
    "var_key": "test",
    "var_value": "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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowConformancePackSolution {

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

```
ICredential auth = new GlobalCredentials()
    .withAk(ak)
    .withSk(sk);

ConfigClient client = ConfigClient.newBuilder()
    .withCredential(auth)
    .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
    .build();
ShowConformancePackRequest request = new ShowConformancePackRequest();
try {
    ShowConformancePackResponse response = client.showConformancePack(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowConformancePackRequest()
        response = client.show_conformance_pack(request)
        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/global"
```

```
config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ShowConformancePackRequest{}  
    response, err := client.ShowConformancePack(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.4 删除合规规则包

### 功能介绍

删除用户的合规规则包。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/resource-manager/domains/{domain\_id}/conformance-packs/  
{conformance\_pack\_id}

表 4-605 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
conformance_pack_id	是	String	合规规则包ID。 最大长度：36

### 请求参数

表 4-606 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

### 响应参数

状态码：400

表 4-607 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-608 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-609 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-610 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeleteConformancePackSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteConformancePackRequest request = new DeleteConformancePackRequest();
        try {
            DeleteConformancePackResponse response = client.deleteConformancePack(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeleteConformancePackRequest()
```



```
response = client.delete_conformance_pack(request)
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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
202	操作接受。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到。

状态码	描述
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.5 列举合规规则包的结果概览

### 功能介绍

列举用户的合规规则包的合规结果概览。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/conformance-packs/compliance/summary

表 4-611 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-612 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

参数	是否必选	参数类型	描述
conformance_pack_name	否	String	合规规则包名称。 最小长度：1 最大长度：64

## 请求参数

表 4-613 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-614 响应 Body 参数

参数	参数类型	描述
value	Array of <b>ConformancePackComplianceSummary</b> objects	合规规则包的合规结果概览列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-615 ConformancePackComplianceSummary

参数	参数类型	描述
id	String	合规规则包ID。
name	String	合规规则包名称。
compliance	String	合规规则包合规结果。

表 4-616 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码：400

表 4-617 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-618 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-619 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

**状态码： 200**

操作成功。

```
{
  "value": [ {
    "id": "a0123456789",
    "name": "conformance_pack_name-e35r",
    "compliance": "NonCompliant"
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class CollectConformancePackComplianceSummarySolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        CollectConformancePackComplianceSummaryRequest request = new
        CollectConformancePackComplianceSummaryRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withConformancePackName("<conformance_pack_name>");
        try {
            CollectConformancePackComplianceSummaryResponse response =
            client.collectConformancePackComplianceSummary(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = CollectConformancePackComplianceSummaryRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.conformance_pack_name = "<conformance_pack_name>"
        response = client.collect_conformance_pack_compliance_summary(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

```
auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CollectConformancePackComplianceSummaryRequest{
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    conformancePackNameRequest:= "<conformance_pack_name>"
    request.ConformancePackName = &conformancePackNameRequest
    response, err := client.CollectConformancePackComplianceSummary(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.6 列举合规规则包的评估结果

### 功能介绍

列举合规规则包的合规规则评估结果。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v1/resource-manager/domains/{domain\_id}/conformance-packs/{conformance\_pack\_id}/compliance

表 4-620 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
conformance_pack_id	是	String	合规规则包ID。 最大长度：36

表 4-621 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
policy_assignment_name	否	String	合规规则名称。 最小长度：1 最大长度：64

## 请求参数

表 4-622 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。



## 响应参数

状态码： 200

表 4-623 响应 Body 参数

参数	参数类型	描述
value	Array of <b>ConformancePackCompliance</b> objects	合规规则包的合规规则评估结果列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-624 ConformancePackCompliance

参数	参数类型	描述
policy_assignment_id	String	合规规则ID。
policy_assignment_name	String	合规规则名称。
policy_assignment_compliance	String	合规规则合规结果。

表 4-625 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-626 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-627 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-628 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-629 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{  
  "value": [ {
```

```
"policy_assignment_id" : "p0123456789",
"policy_assignment_name" : "policy_assignment_name-0709",
"policy_assignment_compliance" : "NonCompliant"
}],
"page_info" : {
  "current_count" : 1,
  "next_marker" : 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListConformancePackComplianceByPackIdSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

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

        ListConformancePackComplianceByPackIdRequest request = new
        ListConformancePackComplianceByPackIdRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withPolicyAssignmentName("<policy_assignment_name>");
        try {
            ListConformancePackComplianceByPackIdResponse response =
            client.listConformancePackComplianceByPackId(request);
            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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkconfig.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = GlobalCredentials(ak, sk) \  
  
    client = ConfigClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListConformancePackComplianceByPackIdRequest()  
        request.limit = <limit>  
        request.marker = "<marker>"  
        request.policy_assignment_name = "<policy_assignment_name>"  
        response = client.list_conformance_pack_compliance_by_pack_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/global"  
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().
```

```
WithRegion(region.ValueOf("<YOUR REGION>")).
WithCredential(auth).
Build()

request := &model.ListConformancePackComplianceByPackIdRequest{}
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
policyAssignmentNameRequest:= "<policy_assignment_name>"
request.PolicyAssignmentName = &policyAssignmentNameRequest
response, err := client.ListConformancePackComplianceByPackId(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.7 列举合规规则包的评估结果详情

### 功能介绍

列举合规规则包的合规规则评估结果详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/conformance-packs/  
{conformance\_pack\_id}/compliance/details

表 4-630 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36
conformance_pack_id	是	String	合规规则包ID。 最大长度：36

表 4-631 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
policy_assignment_name	否	String	合规规则名称。 最小长度：1 最大长度：64

## 请求参数

表 4-632 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-633 响应 Body 参数

参数	参数类型	描述
value	Array of <b>ConformancePackComplianceDetail</b> objects	合规规则包的合规规则评估结果详情列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-634 ConformancePackComplianceDetail

参数	参数类型	描述
policy_assignment_id	String	合规规则ID。
policy_assignment_name	String	合规规则名称。
resource_id	String	评估资源ID。
compliance_state	String	合规规则合规结果。
evaluation_time	String	资源评估时间。

表 4-635 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-636 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-637 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-638 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-639 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{  
  "value": [ {
```



```
"policy_assignment_id" : "p0123456789",
"policy_assignment_name" : "policy_assignment_name-0709",
"resource_id" : "ecs_001",
"compliance_state" : "NonCompliant",
"evaluation_time" : 1667378537566
}],
"page_info" : {
  "current_count" : 1,
  "next_marker" : 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListConformancePackComplianceDetailsByPackIdSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

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

        ListConformancePackComplianceDetailsByPackIdRequest request = new
        ListConformancePackComplianceDetailsByPackIdRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withPolicyAssignmentName("<policy_assignment_name>");
        try {
            ListConformancePackComplianceDetailsByPackIdResponse response =
            client.listConformancePackComplianceDetailsByPackId(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkconfig.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = GlobalCredentials(ak, sk) \  
  
    client = ConfigClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListConformancePackComplianceDetailsByPackIdRequest()  
        request.limit = <limit>  
        request.marker = "<marker>"  
        request.policy_assignment_name = "<policy_assignment_name>"  
        response = client.list_conformance_pack_compliance_details_by_pack_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/global"  
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  

```

```
config.ConfigClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build()

request := &model.ListConformancePackComplianceDetailsByPackIdRequest{}
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
policyAssignmentNameRequest:= "<policy_assignment_name>"
request.PolicyAssignmentName = &policyAssignmentNameRequest
response, err := client.ListConformancePackComplianceDetailsByPackId(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.8 列举合规规则包分数

### 功能介绍

列举用户的合规规则包分数。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/domains/{domain\_id}/conformance-packs/scores

表 4-640 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最大长度：36

表 4-641 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
conformance_pack_name	否	String	合规规则包名称。 最小长度：1 最大长度：64

## 请求参数

表 4-642 请求 Header 参数

参数	是否必选	参数类型	描述
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-643 响应 Body 参数

参数	参数类型	描述
value	Array of <b>ConformancePackScore</b> objects	合规规则包分数查询列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-644 ConformancePackScore

参数	参数类型	描述
id	String	合规规则包ID。
name	String	合规规则包名称。
score	String	合规规则包分数。

表 4-645 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-646 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-647 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-648 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-649 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "value": [{
    "id": "a0123456789",
    "name": "conformance_pack_name-e35r",
    "score": "98.20"
  }],
  "page_info": {
    "current_count": 1,
    "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListConformancePackComplianceScoresSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListConformancePackComplianceScoresRequest request = new
        ListConformancePackComplianceScoresRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withConformancePackName("<conformance_pack_name>");
        try {
            ListConformancePackComplianceScoresResponse response =
            client.listConformancePackComplianceScores(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = ListConformancePackComplianceScoresRequest()
    request.limit = <limit>
    request.marker = "<marker>"
    request.conformance_pack_name = "<conformance_pack_name>"
    response = client.list_conformance_pack_compliance_scores(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListConformancePackComplianceScoresRequest{
        limitRequest:= int32(<limit>)
        request.Limit = &limitRequest
        markerRequest:= "<marker>"
        request.Marker = &markerRequest
        conformancePackNameRequest:= "<conformance_pack_name>"
        request.ConformancePackName = &conformancePackNameRequest
        response, err := client.ListConformancePackComplianceScores(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```



```
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.9 列举预定义合规规则包模板

### 功能介绍

列举预定义的合规规则包的模板。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/conformance-packs/templates

表 4-650 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200

参数	是否必选	参数类型	描述
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
template_key	否	String	预定义合规包模板名称。 最小长度：1 最大长度：128

## 请求参数

表 4-651 请求 Header 参数

参数	是否必选	参数类型	描述
X-Language	否	String	选择接口返回的信息的语言，默认为"zh-cn"中文，可选值：zh-cn和en-us。 缺省值：zh-cn
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

## 响应参数

状态码：200

表 4-652 响应 Body 参数

参数	参数类型	描述
value	Array of <b>Conformance PackTemplate</b> objects	预定义合规规则包模板列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-653 ConformancePackTemplate

参数	参数类型	描述
id	String	预定义合规包模板ID。
template_key	String	预定义合规包模板名称。
description	String	预定义合规包模板描述。
template_body	String	预定义合规包模板内容。
parameters	Map<String,TemplateParameterDefinition>	预定义合规包模板参数。

表 4-654 TemplateParameterDefinition

参数	参数类型	描述
name	String	预定义合规包模板参数名字。
description	String	预定义合规包模板参数描述。
default_value	Object	预定义合规包模板参数默认值。
allowed_values	Array of objects	预定义合规包模板参数允许值列表。
minimum	Float	策略参数的最小值，当参数类型为Integer或Float时生效。
maximum	Float	策略参数的最大值，当参数类型为Integer或Float时生效。
min_items	Integer	策略参数的最小项数，当参数类型为Array时生效。
max_items	Integer	策略参数的最大项数，当参数类型为Array时生效。
min_length	Integer	策略参数的最小字符串长度或每项的最小字符串长度，当参数类型为String或Array时生效。
max_length	Integer	策略参数的最大字符串长度或每项的最大字符串长度，当参数类型为String或Array时生效。
pattern	String	策略参数的字符串正则要求或每项的字符串正则要求，当参数类型为String或Array时生效。
type	String	预定义合规包模板参数类型。

表 4-655 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码：400

表 4-656 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-657 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-658 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

**状态码： 200**

操作成功。

```
{
  "value": [ {
    "id": "k0123456789",
    "template_key": "example-001.yaml",
    "description": "预定义合规包模板示例。",
    "template_body": "Omit it.",
    "parameters": {
      "listOfAllowedFlavors": {
        "description": "The list of allowed flavor types.",
        "type": "Array",
        "default_value": "STAT",
        "allowed_values": [ "STAT" ]
      }
    }
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListBuiltInConformancePackTemplatesSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBuiltInConformancePackTemplatesRequest request = new
        ListBuiltInConformancePackTemplatesRequest();
        request.withLimit(<limit>);
    }
}
```

```
request.withMarker("<marker>");
request.withTemplateKey("<template_key>");
try {
    ListBuiltInConformancePackTemplatesResponse response =
client.listBuiltInConformancePackTemplates(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListBuiltInConformancePackTemplatesRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.template_key = "<template_key>"
        response = client.list_built_in_conformance_pack_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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)
```

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBuiltinConformancePackTemplatesRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    templateKeyRequest:= "<template_key>"
    request.TemplateKey = &templateKeyRequest
    response, err := client.ListBuiltinConformancePackTemplates(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.10 查看预定义合规规则包模板

### 功能介绍

根据ID获取单个预定义合规规则包模板。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/conformance-packs/templates/{template\_id}

表 4-659 路径参数

参数	是否必选	参数类型	描述
template_id	是	String	合规规则包模板ID。 最大长度：36

### 请求参数

表 4-660 请求 Header 参数

参数	是否必选	参数类型	描述
X-Language	否	String	选择接口返回的信息的语言，默认为"zh-cn"中文，可选值：zh-cn和en-us。 缺省值：zh-cn
X-Security-Token	否	String	如果正在使用临时安全凭据，则此header是必需的，该值是临时安全凭据的安全令牌（会话令牌）。

### 响应参数

状态码：200

表 4-661 响应 Body 参数

参数	参数类型	描述
id	String	预定义合规包模板ID。
template_key	String	预定义合规包模板名称。



参数	参数类型	描述
description	String	预定义合规包模板描述。
template_body	String	预定义合规包模板内容。
parameters	Map<String, TemplateParameterDefinition>	预定义合规包模板参数。

表 4-662 TemplateParameterDefinition

参数	参数类型	描述
name	String	预定义合规包模板参数名字。
description	String	预定义合规包模板参数描述。
default_value	Object	预定义合规包模板参数默认值。
allowed_values	Array of objects	预定义合规包模板参数允许值列表。
minimum	Float	策略参数的最小值，当参数类型为Integer或Float时生效。
maximum	Float	策略参数的最大值，当参数类型为Integer或Float时生效。
min_items	Integer	策略参数的最小项数，当参数类型为Array时生效。
max_items	Integer	策略参数的最大项数，当参数类型为Array时生效。
min_length	Integer	策略参数的最小字符串长度或每项的最小字符串长度，当参数类型为String或Array时生效。
max_length	Integer	策略参数的最大字符串长度或每项的最大字符串长度，当参数类型为String或Array时生效。
pattern	String	策略参数的字符串正则要求或每项的字符串正则要求，当参数类型为String或Array时生效。
type	String	预定义合规包模板参数类型。

状态码： 400

表 4-663 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-664 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-665 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-666 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{  
  "id": "k0123456789",
```

```
"template_key": "example-001.yaml",
"description": "预定义合规包模板示例。",
"template_body": "Omit it.",
"parameters": {
  "listOfAllowedFlavors": {
    "description": "The list of allowed flavor types.",
    "type": "Array",
    "default_value": "STAT",
    "allowed_values": [ "STAT" ]
  }
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowBuiltInConformancePackTemplateSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowBuiltInConformancePackTemplateRequest request = new
        ShowBuiltInConformancePackTemplateRequest();
        try {
            ShowBuiltInConformancePackTemplateResponse response =
            client.showBuiltInConformancePackTemplate(request);
            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  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkconfig.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = GlobalCredentials(ak, sk) \  
  
    client = ConfigClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ShowBuiltInConformancePackTemplateRequest()  
        response = client.show_built_in_conformance_pack_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/global"  
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())
```

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.11 创建组织合规规则包

### 功能介绍

创建新的组织合规规则包。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/resource-manager/organizations/{organization\_id}/conformance-packs

表 4-667 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

## 请求参数

表 4-668 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	组织合规规则包名称。 最小长度：1 最大长度：64
excluded_accounts	否	Array of strings	排除配置合规包的账号。
template_key	否	String	预定义合规规则包模板名称。 最大长度：128
template_body	否	String	自定义合规规则包内容。 最小长度：0 最大长度：51200
template_uri	否	String	合规规则包模板OBS地址。 最小长度：0 最大长度：1024
vars_structure	否	Array of <b>VarsStructure</b> objects	合规规则包参数。 数组长度：0 - 50

表 4-669 VarsStructure

参数	是否必选	参数类型	描述
var_key	否	String	参数名称。 最小长度：1 最大长度：64
var_value	否	Object	参数的值。

## 响应参数

状态码：200

表 4-670 响应 Body 参数

参数	参数类型	描述
org_comformance_pack_id	String	组织合规规则包ID。

参数	参数类型	描述
org_conformance_pack_name	String	组织合规规则包名称。
owner_id	String	组织合规规则包创建者。
organization_id	String	组织ID
org_conformance_pack_urn	String	组织合规规则包资源唯一标识。
vars_structure	Array of <b>VarsStructure</b> objects	合规规则包参数。
created_at	String	组织合规规则包创建时间。
updated_at	String	组织合规规则包更新时间。

表 4-671 VarsStructure

参数	参数类型	描述
var_key	String	参数名称。 最小长度：1 最大长度：64
var_value	Object	参数的值。

状态码：400

表 4-672 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-673 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	错误消息内容。

状态码： 500

表 4-674 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

```
POST https://{endpoint}/v1/resource-manager/organizations/{organization_id}/conformance-packs
{
  "name": "org_conformance_pack_name-e35r",
  "template_key": "example-001.yaml",
  "template_body": null,
  "template_uri": null,
  "vars_structure": null
}
```

## 响应示例

状态码： 200

操作成功。

```
{
  "org_conformance_pack_id": "d02b7fa9d5a74e638c1402d0868f71fd",
  "org_conformance_pack_name": "org_conformance_pack_name-e35r",
  "owner_id": "a1b2c043fab784a45ad88f5ef6a4bdaas",
  "organization_id": "o-doxbpkzqsdd51ti1k27w2y8fitakrkn",
  "org_conformance_pack_urn": "rms::a1b2c043fab784a45ad88f5ef6a4bdaas:organizationConformancePack:o-doxbpkzqsdd51ti1k27w2y8fitakrkn/d02b7fa9d5a74e638c1402d0868f71fd",
  "created_at": "2023-02-28T09:39:19Z",
  "updated_at": "2023-02-28T09:39:19Z"
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```



```
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class CreateOrganizationConformancePackSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateOrganizationConformancePackRequest request = new
        CreateOrganizationConformancePackRequest();
        OrgConformancePackRequestBody body = new OrgConformancePackRequestBody();
        body.withTemplateKey("example-001.yaml");
        body.withName("org_conformance_pack_name-e35r");
        request.withBody(body);
        try {
            CreateOrganizationConformancePackResponse response =
            client.createOrganizationConformancePack(request);
            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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

    client = ConfigClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(ConfigRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = CreateOrganizationConformancePackRequest()  
    request.body = OrgConformancePackRequestBody(  
        template_key="example-001.yaml",  
        name="org_conformance_pack_name-e35r"  
    )  
    response = client.create_organization_conformance_pack(request)  
    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/global"  
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := global.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := config.NewConfigClient(  
        config.ConfigClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.CreateOrganizationConformancePackRequest{}  
    templateKeyOrgConformancePackRequestBody := "example-001.yaml"  
    request.Body = &model.OrgConformancePackRequestBody{  
        TemplateKey: &templateKeyOrgConformancePackRequestBody,  
        Name: "org_conformance_pack_name-e35r",  
    }  
    response, err := client.CreateOrganizationConformancePack(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.12 列举组织合规规则包

### 功能介绍

列举用户的组织合规规则包。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/organizations/{organization\_id}/conformance-packs

表 4-675 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

表 4-676 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200

参数	是否必选	参数类型	描述
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
organization_conformance_pack_id	否	String	组织合规规则包ID。 最大长度：36
conformance_pack_name	否	String	合规规则包名称。 最小长度：1 最大长度：64

## 请求参数

无

## 响应参数

状态码：200

表 4-677 响应 Body 参数

参数	参数类型	描述
organization_conformance_packs	Array of <b>OrgConformancePackResponse</b> objects	组织合规规则包查询列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-678 OrgConformancePackResponse

参数	参数类型	描述
org_conformance_pack_id	String	组织合规规则包ID。
org_conformance_pack_name	String	组织合规规则包名称。
owner_id	String	组织合规规则包创建者。

参数	参数类型	描述
organization_id	String	组织ID
org_conformance_pack_urn	String	组织合规规则包资源唯一标识。
vars_structure	Array of <b>VarsStructure</b> objects	合规规则包参数。
created_at	String	组织合规规则包创建时间。
updated_at	String	组织合规规则包更新时间。

表 4-679 VarsStructure

参数	参数类型	描述
var_key	String	参数名称。 最小长度：1 最大长度：64
var_value	Object	参数的值。

表 4-680 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值：0 最大值：200
next_marker	String	下一页的marker。 最小长度：4 最大长度：400

状态码：400

表 4-681 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-682 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-683 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "organization_conformance_packs": [ {
    "org_conformance_pack_id": "d02b7fa9d5a74e638c1402d0868f71fd",
    "org_conformance_pack_name": "org_conformance_pack_name-e35r",
    "owner_id": "a1b2c043fab784a45ad88f5ef6a4bdaas",
    "organization_id": "o-doxbpkzqsdd51ti1k27w2y8fitakrkn",
    "org_conformance_pack_urn":
    "rms::a1b2c043fab784a45ad88f5ef6a4bdaas:organizationConformancePack:o-
    doxbpkzqsdd51ti1k27w2y8fitakrkn/d02b7fa9d5a74e638c1402d0868f71fd",
    "created_at": "2023-02-28T09:39:19Z",
    "updated_at": "2023-02-28T09:39:19Z"
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ListOrganizationConformancePacksSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ListOrganizationConformancePacksRequest request = new
        ListOrganizationConformancePacksRequest();
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        request.withOrganizationConformancePackId("<organization_conformance_pack_id>");
        request.withConformancePackName("<conformance_pack_name>");
        try {
            ListOrganizationConformancePacksResponse response =
            client.listOrganizationConformancePacks(request);
            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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

```
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = GlobalCredentials(ak, sk) \

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

try:
    request = ListOrganizationConformancePacksRequest()
    request.limit = <limit>
    request.marker = "<marker>"
    request.organization_conformance_pack_id = "<organization_conformance_pack_id>"
    request.conformance_pack_name = "<conformance_pack_name>"
    response = client.list_organization_conformance_packs(request)
    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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListOrganizationConformancePacksRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    organizationConformancePackIdRequest := "<organization_conformance_pack_id>"
    request.OrganizationConformancePackId = &organizationConformancePackIdRequest
    conformancePackNameRequest := "<conformance_pack_name>"
    request.ConformancePackName = &conformancePackNameRequest
    response, err := client.ListOrganizationConformancePacks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```



```
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.13 查看组织合规规则包

### 功能介绍

根据ID获取单个组织合规规则包详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/organizations/{organization\_id}/conformance-packs/{conformance\_pack\_id}

表 4-684 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34
conformance_pack_id	是	String	合规规则包ID。 最大长度：36

## 请求参数

无

## 响应参数

状态码： 200

表 4-685 响应 Body 参数

参数	参数类型	描述
org_conformance_pack_id	String	组织合规规则包ID。
org_conformance_pack_name	String	组织合规规则包名称。
owner_id	String	组织合规规则包创建者。
organization_id	String	组织ID
org_conformance_pack_urn	String	组织合规规则包资源唯一标识。
excluded_accounts	Array of strings	排除配置合规包的账号。
vars_structure	Array of <b>VarsStructure</b> objects	合规规则包参数。
created_at	String	组织合规规则包创建时间。
updated_at	String	组织合规规则包更新时间。

表 4-686 VarsStructure

参数	参数类型	描述
var_key	String	参数名称。 最小长度： 1 最大长度： 64
var_value	Object	参数的值。

状态码： 400

表 4-687 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-688 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-689 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-690 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{  
  "org_conformance_pack_id" : "d02b7fa9d5a74e638c1402d0868f71fd",
```

```
"org_conformance_pack_name" : "org_conformance_pack_name-e35r",
"owner_id" : "a1b2c043fab784a45ad88f5ef6a4bdaas",
"organization_id" : "o-doxbpkzqsdd51ti1k27w2y8fitakrkn",
"org_conformance_pack_urn" :
"rms::a1b2c043fab784a45ad88f5ef6a4bdaas:organizationConformancePack:o-
doxbpkzqsdd51ti1k27w2y8fitakrkn/d02b7fa9d5a74e638c1402d0868f71fd",
"created_at" : "2023-02-28T09:39:19Z",
"updated_at" : "2023-02-28T09:39:19Z"
}
```

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowOrganizationConformancePackSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowOrganizationConformancePackRequest request = new
        ShowOrganizationConformancePackRequest();
        try {
            ShowOrganizationConformancePackResponse response =
            client.showOrganizationConformancePack(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowOrganizationConformancePackRequest()
        response = client.show_organization_conformance_pack(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowOrganizationConformancePackRequest{}
    response, err := client.ShowOrganizationConformancePack(request)
```

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

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.14 删除组织合规规则包

### 功能介绍

删除用户的组织合规规则包。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v1/resource-manager/organizations/{organization\_id}/conformance-packs/{conformance\_pack\_id}

表 4-691 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

参数	是否必选	参数类型	描述
conformance_pack_id	是	String	合规规则包ID。 最大长度：36

## 请求参数

无

## 响应参数

状态码：400

表 4-692 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：403

表 4-693 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：404

表 4-694 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码：500

表 4-695 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class DeleteOrganizationConformancePackSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteOrganizationConformancePackRequest request = new
        DeleteOrganizationConformancePackRequest();
        try {
            DeleteOrganizationConformancePackResponse response =
            client.deleteOrganizationConformancePack(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```



```
        e.printStackTrace();
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = DeleteOrganizationConformancePackRequest()
        response = client.delete_organization_conformance_pack(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
```

```
WithSk(sk).
Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
202	操作接受。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.15 查看组织合规规则包部署状态

### 功能介绍

列举用户的组织合规规则包部署状态。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/organizations/{organization\_id}/conformance-packs/statuses

表 4-696 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

表 4-697 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400
organization_conformance_pack_id	否	String	组织合规规则包ID。 最大长度：36
conformance_pack_name	否	String	合规规则包名称。 最小长度：1 最大长度：64

## 请求参数

无

## 响应参数

状态码：200

表 4-698 响应 Body 参数

参数	参数类型	描述
statuses	Array of <b>OrgConformancePackStatus</b> objects	组织合规规则包查询列表。

参数	参数类型	描述
page_info	<b>PageInfo</b> object	分页对象。

表 4-699 OrgConformancePackStatus

参数	参数类型	描述
org_conformance_pack_id	String	组织合规规则包ID。
org_conformance_pack_name	String	组织合规规则包名称。
state	String	合规规则包部署状态
error_message	String	部署或删除组织合规规则包错误时的错误信息
created_at	String	组织合规规则包创建时间。
updated_at	String	组织合规规则包更新时间。

表 4-700 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码：**400**

表 4-701 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-702 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-703 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-704 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{
  "statuses": [ {
    "org_conformance_pack_id": "d02b7fa9d5a74e638c1402d0868f71fd",
    "org_conformance_pack_name": "org_conformance_pack_name-e35r",
    "state": "CREATE_SUCCESSFUL",
    "error_message": null,
    "created_at": "2023-08-16T03:13:50Z",
    "updated_at": "2023-08-16T03:13:50Z"
  } ],
  "page_info": {
    "current_count": 1,
    "next_marker": 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.GlobalCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;  
import com.huaweicloud.sdk.config.v1.*;  
import com.huaweicloud.sdk.config.v1.model.*;  
  
public class ListOrganizationConformancePackStatusesSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new GlobalCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        ConfigClient client = ConfigClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListOrganizationConformancePackStatusesRequest request = new  
        ListOrganizationConformancePackStatusesRequest();  
        request.withLimit("<limit>");  
        request.withMarker("<marker>");  
        request.withOrganizationConformancePackId("<organization_conformance_pack_id>");  
        request.withConformancePackName("<conformance_pack_name>");  
        try {  
            ListOrganizationConformancePackStatusesResponse response =  
            client.listOrganizationConformancePackStatuses(request);  
            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
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ListOrganizationConformancePackStatusesRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.organization_conformance_pack_id = "<organization_conformance_pack_id>"
        request.conformance_pack_name = "<conformance_pack_name>"
        response = client.list_organization_conformance_pack_statuses(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := config.NewConfigClient(
        config.ConfigClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListOrganizationConformancePackStatusesRequest{}
    limitRequest := int32(<limit>)
```

```
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
organizationConformancePackIdRequest:= "<organization_conformance_pack_id>"
request.OrganizationConformancePackId = &organizationConformancePackIdRequest
conformancePackNameRequest:= "<conformance_pack_name>"
request.ConformancePackName = &conformancePackNameRequest
response, err := client.ListOrganizationConformancePackStatuses(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

## 4.9.16 查看组织合规规则包部署详细状态

### 功能介绍

查看指定组织合规规则包在成员账号中的部署状态详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/resource-manager/organizations/{organization\_id}/conformance-packs/  
detailed-statuses



表 4-705 路径参数

参数	是否必选	参数类型	描述
organization_id	是	String	组织ID。 最大长度：34

表 4-706 Query 参数

参数	是否必选	参数类型	描述
conformance_pack_name	是	String	合规规则包名称。 最小长度：1 最大长度：64
organization_conformance_pack_id	否	String	组织合规规则包ID。 最大长度：36
domain_id	否	String	账号ID。 最大长度：36
state	否	String	部署状态，区分大小写 最大长度：20
limit	否	Integer	最大的返回数量。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页参数，通过上一个请求中返回的marker信息作为输入，获取当前页。 最小长度：4 最大长度：400

## 请求参数

无

## 响应参数

状态码：200

表 4-707 响应 Body 参数

参数	参数类型	描述
statuses	Array of <b>OrgConformancePackDetailedStatus</b> objects	组织合规规则包查询列表。
page_info	<b>PageInfo</b> object	分页对象。

表 4-708 OrgConformancePackDetailedStatus

参数	参数类型	描述
domain_id	String	成员账号ID。
conformance_pack_id	String	合规规则包ID。
conformance_pack_name	String	合规规则包名称。
state	String	合规规则包部署状态
error_message	String	部署或删除组织合规规则包错误时的错误信息
created_at	String	创建时间。
updated_at	String	更新时间。

表 4-709 PageInfo

参数	参数类型	描述
current_count	Integer	当前页的资源数量。 最小值： <b>0</b> 最大值： <b>200</b>
next_marker	String	下一页的marker。 最小长度： <b>4</b> 最大长度： <b>400</b>

状态码： 400

表 4-710 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 403

表 4-711 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 404

表 4-712 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

状态码： 500

表 4-713 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误消息内容。

## 请求示例

无

## 响应示例

状态码： 200

操作成功。

```
{  
  "statuses": [ {
```

```
"domain_id" : "d02b7fa9d5a74e638c1402d0868f71fd",
"conformance_pack_id" : "d02b7fa9d5a74e638c1402d0868f71fd",
"conformance_pack_name" : "org_conformance_pack_name-e35r",
"state" : "CREATE_SUCCESSFUL",
"error_message" : "null",
"created_at" : "2023-01-17T09:38:26.165Z",
"updated_at" : "2023-01-17T09:38:26.165Z"
}],
"page_info" : {
  "current_count" : 1,
  "next_marker" : 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.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.config.v1.region.ConfigRegion;
import com.huaweicloud.sdk.config.v1.*;
import com.huaweicloud.sdk.config.v1.model.*;

public class ShowOrganizationConformancePackDetailedStatusesSolution {

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

        ICredential auth = new GlobalCredentials()
            .withAk(ak)
            .withSk(sk);

        ConfigClient client = ConfigClient.newBuilder()
            .withCredential(auth)
            .withRegion(ConfigRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowOrganizationConformancePackDetailedStatusesRequest request = new
        ShowOrganizationConformancePackDetailedStatusesRequest();
        request.withConformancePackName("<conformance_pack_name>");
        request.withOrganizationConformancePackId("<organization_conformance_pack_id>");

        request.withState(ShowOrganizationConformancePackDetailedStatusesRequest.StateEnum.fromValue("<state>"));
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        try {
            ShowOrganizationConformancePackDetailedStatusesResponse response =
            client.showOrganizationConformancePackDetailedStatuses(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } 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

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkconfig.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

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

    try:
        request = ShowOrganizationConformancePackDetailedStatusesRequest()
        request.conformance_pack_name = "<conformance_pack_name>"
        request.organization_conformance_pack_id = "<organization_conformance_pack_id>"
        request.state = "<state>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.show_organization_conformance_pack_detailed_statuses(request)
        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/global"
    config "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/config/v1/region"
)

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

```
sk := os.Getenv("CLOUD_SDK_SK")

auth := global.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := config.NewConfigClient(
    config.ConfigClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowOrganizationConformancePackDetailedStatusesRequest{}
request.ConformancePackName = "<conformance_pack_name>"
organizationConformancePackIdRequest := "<organization_conformance_pack_id>"
request.OrganizationConformancePackId = &organizationConformancePackIdRequest
stateRequest :=
model.GetShowOrganizationConformancePackDetailedStatusesRequestStateEnum().<STATE>
request.State = &stateRequest
limitRequest := int32(<limit>)
request.Limit = &limitRequest
markerRequest := "<marker>"
request.Marker = &markerRequest
response, err := client.ShowOrganizationConformancePackDetailedStatuses(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	操作成功。
400	输入参数不合法。
403	用户身份认证失败。
404	资源未找到
500	服务器内部错误。

## 错误码

请参见[错误码](#)。

# 5 权限和授权项

## 5.1 权限策略及授权项说明

如果您需要对Config进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用Config的其它功能。

策略是以JSON格式描述权限集的语言。默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略，才能使用户组中的用户获得策略定义的权限，这一过程称为授权。授权后，用户就可以基于策略对云服务进行操作。

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

如果您需要允许或者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资源记录器，那么这个IAM用户被授予的策略中必须包含允许“rms:trackerConfig:get”的授权项，该接口才能调用成功。

## 支持的授权项

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。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](#)。

Config支持的自定义策略授权项如下所示：

**资源清单**，包含资源接口对应的授权项，如查看资源历史、列举指定类型的资源等接口。

**资源记录器**，包含资源记录器接口对应的授权项，如查询资源记录器、创建资源记录器、删除资源记录器等接口。

**资源合规**，包含资源合规接口对应的授权项，如创建合规规则、更新合规规则、删除合规规则等接口。

**高级查询**，包含高级查询接口对应的授权项，如运行ResourceQL查询、创建查询语句、列举查询语句等接口。

**资源聚合器**，包含资源聚合器接口对应的授权项，如创建资源聚合器、更新资源聚合器、删除资源聚合器等接口。

**合规规则包**，包含合规规则包接口对应的授权项，如创建合规规则包、列举合规规则包、删除合规规则包等接口。

#### 📖 说明

“√”表示支持，“x”表示暂不支持。

## 5.2 资源清单

权限	对应API接口	授权项	IAM项目	企业项目
查看资源历史	GET /v1/resource-manager/domains/{domain_id}/resources/{resource_id}/history	rms:resources:getHistory	√	不涉及
查询资源关系	GET /v1/resource-manager/domains/{domain_id}/all-resources/{resource_id}/relations	rms:resources:getRelation	√	不涉及
列举指定类型的资源	GET /v1/resource-manager/domains/{domain_id}/provider/{provider}/type/{type}/resources	rms:resources:list	√	不涉及
查询单个资源	GET /v1/resource-manager/domains/{domain_id}/provider/{provider}/type/{type}/resources/{resource_id}	rms:resources:get	√	不涉及



权限	对应API接口	授权项	IAM项目	企业项目
列举资源记录器收集的全部资源	GET /v1/resource-manager/domains/{domain_id}/tracked-resources	rms:resources:list	√	不涉及
查询资源记录器收集的資源数量	GET /v1/resource-manager/domains/{domain_id}/tracked-resources/count	rms:resources:list	√	不涉及
列举资源记录器收集的資源标签	GET /v1/resource-manager/domains/{domain_id}/tracked-resources/tags	rms:resources:list	√	不涉及
列举资源记录器收集的資源概要	GET /v1/resource-manager/domains/{domain_id}/tracked-resources/summary	rms:resources:list	√	不涉及
查询资源记录器收集的单个資源	GET /v1/resource-manager/domains/{domain_id}/tracked-resources/{resource_id}	rms:resources:get	√	不涉及
列举資源概要	GET /v1/resource-manager/domains/{domain_id}/all-resources/summary	rms:resources:list	√	不涉及
查询資源数量	GET /v1/resource-manager/domains/{domain_id}/all-resources/count	rms:resources:list	√	不涉及
列举所有資源	GET /v1/resource-manager/domains/{domain_id}/all-resources	rms:resources:list	√	不涉及
查询支持的云服务、資源和区域列表	GET /v1/resource-manager/domains/{domain_id}/providers	无需授权	不涉及	不涉及
列举資源标签	GET /v1/resource-manager/domains/{domain_id}/all-resources/tags	rms:resources:list	√	不涉及
查询账号下的单个資源	GET /v1/resource-manager/domains/{domain_id}/all-resources/{resource_id}	rms:resources:get	√	不涉及

## 5.3 资源记录器

权限	对应API接口	授权项	IAM项目	企业项目
查询资源记录器	GET /v1/resource-manager/domains/{domain_id}/tracker-config	rms:trackerConfig:get	√	不涉及
创建或更新资源记录器	PUT /v1/resource-manager/domains/{domain_id}/tracker-config	rms:trackerConfig:put	√	不涉及
删除资源记录器	DELETE /v1/resource-manager/domains/{domain_id}/tracker-config	rms:trackerConfig:delete	√	不涉及

## 5.4 资源合规

权限	对应API接口	授权项	依赖的授权项	IAM项目	企业项目
列出内置策略	GET /v1/resource-manager/policyDefinitions	rms:policyDefinitions:get	-	√	不涉及
查询单个内置策略	GET /v1/resource-manager/policy-definitions/{policy_definition_id}	rms:policyDefinitions:get	-	√	不涉及
更新合规规则	PUT /v1/resource-manager/domains/{domain_id}/policy-assignments/{policy_assignment_id}	rms:policyAssignments:update	-	√	不涉及
启用合规规则	POST /v1/resource-manager/domains/{domain_id}/policy-assignments/{policy_assignment_id}/enable	rms:policyAssignments:update	-	√	不涉及

权限	对应API接口	授权项	依赖的授权项	IAM项目	企业项目
停用合规规则	POST /v1/resource-manager/domains/{domain_id}/policy-assignments/{policy_assignment_id}/disable	rms:policyAssignments:update	-	√	不涉及
创建合规规则	PUT /v1/resource-manager/domains/{domain_id}/policy-assignments	rms:policyAssignments:create	-	√	不涉及
列出合规规则	GET /v1/resource-manager/domains/{domain_id}/policy-assignments	rms:policyAssignments:get	-	√	不涉及
查看单个合规规则	GET /v1/resource-manager/domains/{domain_id}/policy-assignments/{policy_assignment_id}	rms:policyAssignments:get	-	√	不涉及
删除合规规则	DELETE /v1/resource-manager/domains/{domain_id}/policy-assignments/{policy_assignment_id}	rms:policyAssignments:delete	-	√	不涉及
获取规则的评估状态	GET /v1/resource-manager/domains/{domain_id}/policy-states/evaluation-state	rms:policyStates:get	-	√	不涉及
获取资源的合规结果	GET /v1/resource-manager/domains/{domain_id}/resources/{resource_id}/policy-states	rms:policyStates:get	-	√	不涉及
获取用户的合规结果	GET /v1/resource-manager/domains/{domain_id}/policy-states	rms:policyStates:get	-	√	不涉及

权限	对应API接口	授权项	依赖的授权项	IA M 项目	企业 项目
获取规则的合规结果	GET /v1/resource-manager/domains/{domain_id}/policy-assignments/{policy_assignment_id}/policy-states	rms:policyStates:get	-	√	不涉及
运行合规评估	POST /v1/resource-manager/domains/{domain_id}/policy-states/run-evaluation	rms:policyStates:runEvaluation	-	√	不涉及
更新合规评估结果	GET /v1/resource-manager/domains/{domain_id}/policy-states	rms:policyStates:update	-	√	不涉及
创建或更新组织合规规则	PUT /v1/resource-manager/organizations/{organization_id}/policy-assignments	rms:organizationPolicyAssignments:put	<ul style="list-style-type: none"> <li>• organizations:organizations:get</li> <li>• organizations:accounts:list</li> <li>• organizations:delegatedAdministrators:list</li> <li>• organizations:trustedServices:enable</li> <li>• organizations:trustedServices:list</li> </ul>	√	不涉及
查询组织合规规则列表	GET /v1/resource-manager/organizations/{organization_id}/policy-assignments	rms:organizationPolicyAssignments:list	organizations:organizations:get	√	不涉及

权限	对应API接口	授权项	依赖的授权项	IAM项目	企业项目
查询指定组织合规规则	GET /v1/resource-manager/organizations/{organization_id}/policy-assignments/{organization_policy_assignment_id}	rms:organizationPolicyAssignments:get	organizations:organizations:get	√	不涉及
查询组织合规规则部署状态	GET /v1/resource-manager/organizations/{organization_id}/policy-assignment-statuses	rms:organizationPolicyAssignments:list	organizations:organizations:get	√	不涉及
查询组织内每个成员账号合规规则部署的详细状态	GET /v1/resource-manager/organizations/{organization_id}/policy-assignment-detailed-status	rms:organizationPolicyAssignments:list	organizations:organizations:get	√	不涉及
删除组织合规规则	DELETE /v1/resource-manager/organizations/{organization_id}/policy-assignments/{organization_policy_assignment_id}	rms:organizationPolicyAssignments:delete	organizations:organizations:get	√	不涉及

## 5.5 高级查询

权限	对应API接口	授权项	IAM项目	企业项目
运行高级查询	POST /v1/resource-manager/domains/{domain_id}/run-query	rms:resources:runQuery	√	不涉及
创建查询语句	POST /v1/resource-manager/domains/{domain_id}/stored-queries	rms:storedQueries:create	√	不涉及
查看查询语句	GET /v1/resource-manager/domains/{domain_id}/stored-queries/{query_id}	rms:storedQueries:get	√	不涉及

权限	对应API接口	授权项	IAM项目	企业项目
列举查询语句	GET /v1/resource-manager/domains/{domain_id}/stored-queries	rms:storedQueries:list	√	不涉及
更新查询语句	PUT /v1/resource-manager/domains/{domain_id}/stored-queries/{query_id}	rms:storedQueries:update	√	不涉及
删除查询语句	DELETE /v1/resource-manager/domains/{domain_id}/stored-queries/{query_id}	rms:storedQueries:delete	√	不涉及
列举高级查询 Schemas	GET /v1/resource-manager/domains/{domain_id}/schemas	rms:schemas:list	√	不涉及

## 5.6 资源聚合器

权限	对应API接口	授权项	IAM项目	企业项目
创建资源聚合器授权	PUT /v1/resource-manager/domains/{domain_id}/aggregators/aggregation-authorization	rms:aggregationAuthorizations:create	√	不涉及
删除资源聚合器授权	DELETE /v1/resource-manager/domains/{domain_id}/aggregators/aggregation-authorization/{authorized_account_id}	rms:aggregationAuthorizations:delete	√	不涉及
查询资源聚合器授权列表	GET /v1/resource-manager/domains/{domain_id}/aggregators/aggregation-authorization	rms:aggregationAuthorizations:list	√	不涉及
创建资源聚合器	PUT /v1/resource-manager/domains/{domain_id}/aggregators	rms:aggregators:create	√	不涉及
查询资源聚合器列表	GET /v1/resource-manager/domains/{domain_id}/aggregators	rms:aggregators:list	√	不涉及
查询指定资源聚合器详情	GET /v1/resource-manager/domains/{domain_id}/aggregators/{aggregator_id}	rms:aggregators:get	√	不涉及

权限	对应API接口	授权项	IAM项目	企业项目
查询指定资源聚合器聚合账号的状态信息	GET /v1/resource-manager/domains/{domain_id}/aggregators/{aggregator_id}/aggregator-sources-status	rms:aggregators:get	√	不涉及
删除资源聚合器	DELETE /v1/resource-manager/domains/{domain_id}/aggregators/{aggregator_id}	rms:aggregators:delete	√	不涉及
更新资源聚合器	PUT /v1/resource-manager/domains/{domain_id}/aggregators/{aggregator_id}	rms:aggregators:update	√	不涉及
删除聚合器账号中挂起的授权请求	DELETE /v1/resource-manager/domains/{domain_id}/aggregators/pending-aggregation-request/{requester_account_id}	rms:aggregationRequests:delete	√	不涉及
查询所有挂起的聚合请求列表	GET /v1/resource-manager/domains/{domain_id}/aggregators/pending-aggregation-request	rms:aggregationRequests:list	√	不涉及
查询聚合器中资源的列表	POST /v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/aggregate-discovered-resources	rms:aggregatorResources:list	√	不涉及
对指定聚合器执行高级查询	POST /v1/resource-manager/domains/{domain_id}/aggregators/{aggregator_id}/run-query	rms:aggregatorResources:runQuery	√	不涉及
查询源账号中资源的详情	POST /v1/resource-manager/domains/{domain_id}/aggregators/aggregate-resource-config	rms:aggregatorResources:get	√	不涉及
查询聚合器中账号资源的计数	POST /v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/aggregate-discovered-resource-counts	rms:aggregatorResources:list	√	不涉及

权限	对应API接口	授权项	IAM项目	企业项目
查询聚合器中一个或多个账号的合规概况	POST /v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/policy-states/compliance-summary	rms:aggregatorResources:list	√	不涉及
查询聚合合规规则列表	POST /v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/policy-assignments/compliance	rms:aggregatorResources:list	√	不涉及
查询指定聚合合规规则的资源评估结果详情	POST /v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/policy-states/compliance-details	rms:aggregatorResources:list	√	不涉及
查询指定聚合合规规则详情	POST /v1/resource-manager/domains/{domain_id}/aggregators/aggregate-data/policy-assignment/detail	rms:aggregatorResources:get	√	不涉及

## 5.7 合规规则包

权限	对应API接口	授权项	依赖的授权项	IAM项目	企业项目
创建合规规则包	POST /v1/resource-manager/domains/{domain_id}/conformance-packs	rms:conformancePacks:create	<ul style="list-style-type: none"> <li>rf:stack:createStack</li> <li>rf:stack:getStackMetadata</li> <li>rf:stack:listStackResources</li> </ul>	√	不涉及
查看合规规则包	GET /v1/resource-manager/domains/{domain_id}/conformance-packs/{conformance_pack_id}	rms:conformancePacks:get	-	√	不涉及



权限	对应API接口	授权项	依赖的授权项	IA M 项目	企业 项目
列举合规规则包	GET /v1/resource-manager/domains/{domain_id}/conformance-packs	rms:conformancePacks:list	-	√	不涉及
删除合规规则包	DELETE /v1/resource-manager/domains/{domain_id}/conformance-packs/{conformance_pack_id}	rms:conformancePacks:delete	<ul style="list-style-type: none"> <li>rf:stack:deleteStack</li> <li>rf:stack:getStackMetadata</li> </ul>	√	不涉及
列举合规规则包的结果概览	GET /v1/resource-manager/domains/{domain_id}/conformance-packs/compliance/summary	rms:conformancePacks:list	-	√	不涉及
列举合规规则包的评估结果	GET /v1/resource-manager/domains/{domain_id}/conformance-packs/{conformance_pack_id}/compliance	rms:conformancePacks:get	-	√	不涉及
列举合规规则包的评估结果详情	GET /v1/resource-manager/domains/{domain_id}/conformance-packs/{conformance_pack_id}/compliance/details	rms:conformancePacks:get	-	√	不涉及
列举合规规则包分数	GET /v1/resource-manager/domains/{domain_id}/conformance-packs/scores	rms:conformancePacks:list	-	√	不涉及

权限	对应API接口	授权项	依赖的授权项	IA M 项目	企业 项目
创建组织 合规规则 包	POST /v1/resource- manager/ organizations/ {organization_id}/ conformance-packs	rms:organizati onConforman cePacks:create	<ul style="list-style-type: none"> <li>• organizatio ns:organiza tions:get</li> <li>• organizatio ns:account s:list</li> <li>• organizatio ns:delegat edAdministr ators:list</li> <li>• organizatio ns:trustedS ervices:ena ble</li> <li>• organizatio ns:trustedS ervices:list</li> </ul>	√	不涉 及
列举组织 合规规则 包	GET /v1/resource- manager/ organizations/ {organization_id}/ conformance-packs	rms:organizati onConforman cePacks:list	organizations: organizations: get	√	不涉 及
查看组织 合规规则 包	GET /v1/resource- manager/ organizations/ {organization_id}/ conformance-packs/ {conformance_pack_id }	rms:organizati onConforman cePacks:get	organizations: organizations: get	√	不涉 及
删除组织 合规规则 包	DELETE /v1/resource- manager/ organizations/ {organization_id}/ conformance-packs/ {conformance_pack_id }	rms:organizati onConforman cePacks:delete	organizations: organizations: get	√	不涉 及
查看组织 合规规则 包部署状 态	GET /v1/resource- manager/ organizations/ {organization_id}/ conformance-packs/ statuses	rms:organizati onConforman cePacks:list	organizations: organizations: get	√	不涉 及

权限	对应API接口	授权项	依赖的授权项	IA M 项目	企业 项目
查看组织 合规规则 包部署详 细状态	GET /v1/resource- manager/ organizations/ {organization_id}/ conformance-packs/ detailed-statuses	rms:organizati onConforman cePacks:list	organizations: organizations: get	√	不涉 及

# 6 附录

## 6.1 错误码

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。

更多服务错误码请参见[API错误中心](#)。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010001	some region is restricted to access.	区域非法。	请检查请求参数。
400	RMS.00010002	resource type invalid.	资源类型非法。	请检查请求参数。
400	RMS.00010003	service invalid.	服务非法。	请检查请求参数。
400	RMS.00010004	invalid parameter.	不合法的参数。	请检查请求参数。
400	RMS.00010007	The max length of domainId must be less than or equal to 36.	domainId的长度必须小于等于36。	请检查账号ID是否正确。
400	RMS.00010008	domainId is illegal.	domainId非法。	请检查账号ID是否正确。
400	RMS.00010009	The max length of id must be less than or equal to 512.	资源ID的长度必须小于等于512。	请检查请求参数。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010010	id is illegal.	资源ID非法。	请检查请求参数。
400	RMS.00010011	The max length of name must be less than or equal to 256.	资源名称的长度必须小于等于256。	请检查请求参数。
400	RMS.00010012	name is illegal.	资源名称非法。	请检查请求参数。
400	RMS.00010013	The size of resource type array must be between 1 and 100.	资源类型列表的长度必须介于1和100之间。	请检查请求参数。
400	RMS.00010014	The max length of resource type must be less than or equal to 40.	资源类型的长度必须小于等于40。	请检查请求参数。
400	RMS.00010015	resource type is illegal.	资源类型非法。	请检查请求参数。
400	RMS.00010016	The size of region id array must be between 1 and 10.	区域ID列表的长度必须介于1和10之间。	请检查请求参数。
400	RMS.00010017	The max length of region id must be less than or equal to 36.	区域ID的长度必须小于等于36。	请检查请求参数。
400	RMS.00010018	region id is illegal.	区域ID非法。	请检查请求参数。
400	RMS.00010019	The size of enterprise project array must be between 1 and 10.	企业项目列表的长度必须介于1和10之间。	请检查请求参数。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010020	enterprise project is invalid.	无效的企业项目。	请检查请求参数。
400	RMS.00010021	The max length of project id array must be between 1 and 10.	项目ID列表的长度必须介于1和10之间。	请检查请求参数。
400	RMS.00010022	The max length of project id must be less than or equal to 36.	项目ID的长度必须小于等于36。	请检查请求参数。
400	RMS.00010023	project id is illegal.	项目ID非法。	请检查请求参数。
400	RMS.00010024	The minimum value of limit must be greater than or equal to 1.	分页限制值必须大于等于1。	请检查请求参数。
400	RMS.00010025	The maximum value of limit must be less than or equal to 200.	分页限制值必须小于等于200。	请检查请求参数。
400	RMS.00010026	The minimum value of offset must be greater than or equal to 1.	偏移值必须大于等于1。	请检查请求参数。
400	RMS.00010027	The maximum value of offset must be less than or equal to 1000.	偏移值必须小于等于1000。	请检查请求参数。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010028	The number of tag keys must be between 1 and 5.	不同标签键的数量必须介于1和5之间。	请检查请求参数。
400	RMS.00010029	The length of a tag key must be less than or equal to 128.	标签的键名长度必须小于等于128。	请检查请求参数。
400	RMS.00010030	The number of values in a tag key must be less than or equal to 5.	同一标签键的键值数量必须小于等于5。	请检查请求参数。
400	RMS.00010031	The length of a tag value must be less than or equal to 255.	标签的键值长度必须小于等于255。	请检查请求参数。
400	RMS.00010032	Tag value is illegal.	标签键值非法。	请检查请求参数。
400	RMS.00010033	Tag key is illegal.	标签键名非法。	请检查请求参数。
400	RMS.00010034	marker is illegal.	标记非法。	请检查请求参数。
400	RMS.00010035	The length of marker must be between 4 and 400.	标记长度必须介于4和400之间。	请检查请求参数。
400	RMS.00010036	The minimum value of limit must be greater than or equal to 1.	分页限制值必须大于等于1。	请检查请求参数。
400	RMS.00010037	The maximum value of limit must be less than or equal to 200.	分页限制值必须小于等于200。	请检查请求参数。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010038	The minimum value of offset must be greater than or equal to 1.	偏移值必须大于等于1。	请检查请求参数。
400	RMS.00010039	The maximum value of offset must be less than or equal to 1000.	偏移值必须小于等于1000。	请检查请求参数。
400	RMS.00010040	The max length of auth action must be less than or equal to 64.	操作的长度必须小于等于64。	请检查请求参数。
400	RMS.00010041	policy assignment id is illegal.	policyAssignmentId非法。	请检查请求参数。
400	RMS.00010042	policy definition id is illegal.	policyDefinitionId非法。	请检查请求参数。
400	RMS.00010043	The max length of policy assignment id must be less than or equal to 36.	policyAssignmentId的长度必须小于等于36。	请检查请求参数。
400	RMS.00010044	The max length of policy definition id must be less than or equal to 36.	policyDefinitionId的长度必须小于等于36。	请检查请求参数。
400	RMS.00010045	The max length of name must be less than or equal to 64.	名称的长度必须小于等于64。	请检查请求参数。



状态码	错误码	错误信息	描述	处理措施
400	RMS.00010046	name is illegal.	名称非法。	请检查请求参数。
400	RMS.00010047	The max length of keyword must be less than or equal to 64.	关键字的长度必须小于等于64。	请检查请求参数。
400	RMS.00010048	keyword is illegal.	关键字非法。	请检查请求参数。
400	RMS.00010049	The max length of compliance state must be less than or equal to 16.	合规状态的长度必须小于等于16。	请检查请求参数。
400	RMS.00010050	compliance state is illegal.	合规状态非法。	请检查请求参数。
400	RMS.00010051	The max length of policy assignment state must be less than or equal to 16.	合规策略状态的长度必须小于等于16。	请检查请求参数。
400	RMS.00010052	policy assignment state is illegal.	合规策略状态非法。	请检查请求参数。
400	RMS.00010053	X-Language header is invalid.	非法的X-Language请求头。	请检查请求参数。
400	RMS.00010054	The max length of queryId must be less than or equal to 36.	QueryId的长度必须小于等于36。	请检查请求参数。
400	RMS.00010055	QueryId is illegal.	QueryId非法。	请检查请求参数。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010056	The minimum value of limit must be greater than or equal to 1.	分页限制值必须大于等于1。	请检查请求参数。
400	RMS.00010057	The maximum value of limit must be less than or equal to 200.	分页限制值必须小于等于200。	请检查请求参数。
400	RMS.00010058	marker is illegal.	marker标记非法。	请检查请求参数。
400	RMS.00010059	The length of marker must be between 4 and 400.	marker标记长度必须介于4和400之间。	请检查请求参数。
400	RMS.00010060	Incorrect resource query language.	错误的高级查询语句。	检查高级查询语句是否正确。
400	RMS.00010061	Stored query name is illegal.	高级查询名称非法。	请检查请求参数。
400	RMS.00010062	The length of stored query name must be between 1 and 64.	高级查询名称长度必须介于1和64之间。	请检查请求参数。
400	RMS.00010064	Organization_id is illegal.	organization_id非法。	请检查请求参数。
400	RMS.00010065	Organization_id length must be between 1 and 34.	organization_id长度必须小于等于34。	请检查请求参数。
400	RMS.00010066	Organization_policy_assignment_id is illegal.	organization_policy_assignment_id非法。	请检查请求参数。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010067	Organization_policy_assignment_id length must be between 1 and 32.	organization_policy_assignment_id长度必须小于等于32。	请检查请求参数。
400	RMS.00010068	Organization_policy_assignment_name is illegal.	organization_policy_assignment_name非法。	请检查请求参数。
400	RMS.00010069	Organization_policy_assignment_name must be between 1 and 64.	organization_policy_assignment_name长度必须小于等于64。	请检查请求参数。
400	RMS.00010070	The aggregator is not found.	聚合器不存在。	请检查请求参数。
400	RMS.00010071	Aggregator_id is illegal.	aggregator_id非法。	请检查请求参数。
400	RMS.00010072	Aggregator_id length must be between 1 and 32.	aggregator_id长度必须小于等于32。	请检查请求参数。
400	RMS.00010073	Authorized_account_id is illegal.	authorized_account_id非法。	请检查请求参数。
400	RMS.00010074	Authorized_account_id length must be between 1 and 36.	authorized_account_id长度必须小于等于36。	请检查请求参数。
400	RMS.00010075	Requester_account_id is illegal.	requester_account_id非法。	请检查请求参数。
400	RMS.00010076	Requester_account_id length must be between 1 and 36.	requester_account_id长度必须小于等于36。	请检查请求参数。
400	RMS.00010077	Aggregator_name is illegal.	aggregator_name非法。	请检查请求参数。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010078	Aggregator_name length must be between 1 and 64.	aggregator_name长度必须小于等于64。	请检查请求参数。
400	RMS.00010080	Aggregator_update_status length must be between 1 and 16.	aggregator_update_status长度必须小于16。	请检查请求参数。
400	RMS.00010081	The maximum value of limit is 2000.	分页限制值必须小于等于2000。	请检查请求参数。
400	RMS.00010082	The number of aggregators exceed quota.	聚合器数量超过配额，组织聚合器上限数量为1，账号聚合器上限数量为30。	请检查资源聚合器剩余配额。
400	RMS.00010083	Conformance_pack_name is illegal.	合规规则包名称非法。	请检查请求参数。
400	RMS.00010084	Conformance_pack_name length must be between 1 and 64.	合规规则包名称长度必须小于等于64。	请检查请求参数。
400	RMS.00010085	Conformance_pack_id is illegal.	合规规则包ID非法。	请检查请求参数。
400	RMS.00010086	Conformance_pack_id length must be between 1 and 36.	合规规则包ID长度必须小于等于36。	请检查请求参数。
400	RMS.00010087	Template_key is illegal.	预定义合规规则包模板的名称非法。	请检查请求参数。
400	RMS.00010088	Template_key length must be between 1 and 128.	预定义合规规则包模板的名称长度必须小于等于128。	请检查请求参数。

状态码	错误码	错误信息	描述	处理措施
400	RMS.00010089	Template_id is illegal.	预定义合规规则包模板的ID非法。	请检查请求参数。
400	RMS.00010090	Template_id length must be between 1 and 36.	预定义合规规则包模板的ID长度必须小于等于36。	请检查请求参数。
400	RMS.00010091	The tracker-config is not enabled.	资源记录器未开启。	请开启资源记录器。
400	RMS.00010092	The number of accounts operated by the aggregator exceeds the limit.	聚合器操作超过限制，账号聚合器七天内增加或删除的最大账户数为1000。	请检查七天内增加或删除的最大账户数是否超过1000。
400	RMS.00010093	Exceeded limit on organization aggregator operations.	聚合器操作超过限制，组织聚合器一天内新增最大数为1。	请检查一天内创建的组织聚合器个数。
401	RMS.00010079	Incorrect IAM authentication information: Authorization header is missing.	IAM认证错误：缺少 Authorization header。	请添加认证 header。
403	RMS.00010006	access denied.	访问拒绝。	请检查权限是否正确。
404	RMS.00010005	resource not found.	资源未找到。	请检查资源是否存在。
429	RMS.00010063	ResourceQL is busy.	高级查询繁忙中。	请稍后重试。
500	RMS.INTERNAL.ERROR	Internal server error occurred.	服务内部错误。	请稍后再试或联系客服。

## 6.2 支持的服务和资源类型

Config支持的云服务（provider）和资源类型（type）可通过如下两种方式获取：

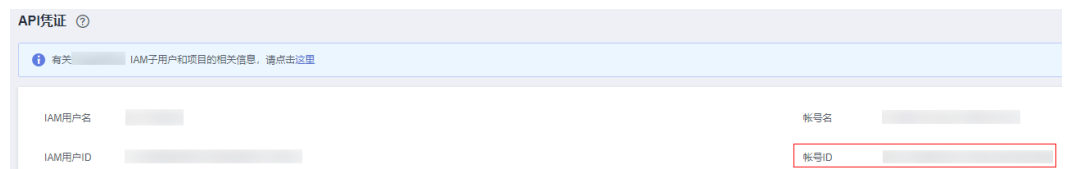
- API方式  
通过调用[列举云服务](#)接口查询Config支持的云服务、资源和区域列表。其中 provider 字段为云服务名称，resource\_types 中的 name 字段为资源类型名称。
- 管理控制台方式  
Config支持的服务和资源类型详见[支持的服务和区域](#)，在列表中“资源类型”列的括号中可获取相关信息。

## 6.3 获取账号 ID

在调用接口的时候，部分URL中需要填入账号ID（domain-id），所以需要先在管理控制台上获取到账号ID。账号ID获取步骤如下：

1. 登录管理控制台。
2. 鼠标悬停在右上角的用户名，选择下拉列表中的“我的凭证”。  
在“API凭证”页面查看账号ID。

图 6-1 获取账号 ID



## 6.4 状态码

- 正常

返回值	说明
200 OK	GET和PUT操作正常返回。
201 Created	POST操作正常返回。
202 Accepted	请求已被接受。
204 No Content	DELETE操作正常返回。

- 异常

返回值	说明
400 Bad Request	服务器未能处理请求。
401 Unauthorized	被请求的页面需要用户名和密码。
403 Forbidden	对被请求页面的访问被禁止。
404 Not Found	服务器无法找到被请求的页面。
405 Method Not Allowed	请求中指定的方法不被允许。

返回值	说明
406 Not Acceptable	服务器生成的响应无法被客户端所接受。
407 Proxy Authentication Required	用户必须首先使用代理服务器进行验证，这样请求才会被处理。
408 Request Timeout	请求超出了服务器的等待时间。
409 Conflict	由于冲突，请求无法被完成。
500 Internal Server Error	请求未完成。服务异常。
501 Not Implemented	请求未完成。服务器不支持所请求的功能。
502 Bad Gateway	请求未完成。服务器从上游服务器收到一个无效的响应。
503 Service Unavailable	请求未完成。系统暂时异常。
504 Gateway Timeout	网关超时。

## 6.5 获取项目 ID

### 操作场景

在调用接口的时候，部分URL中需要填入项目ID，所以需要获取到项目ID。有如下两种获取方式：

- [调用API获取项目ID](#)
- [从控制台获取项目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列表](#)API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从[地区和终端节点](#)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。

```
{
  "projects": [
    {
      "domain_id": "65ewtrgaggshhk1223245sghjlse684b",
      "is_domain": false,
      "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
      "name": "project_name",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4adasfjljaaakla12334jklga9sasfg"
      },
      "id": "a4adasfjljaaakla12334jklga9sasfg",
      "enabled": true
    }
  ],
}
```

```
"links": {  
  "next": null,  
  "previous": null,  
  "self": "https://www.example.com/v3/projects"  
}
```

## 从控制台获取项目 ID

从控制台获取项目ID的步骤如下：

1. 登录管理控制台。
2. 鼠标悬停在右上角的用户名，选择下拉列表中的“我的凭证”。  
在“API凭证”页面的项目列表中查看项目ID。

图 6-2 查看项目 ID





# 7 修订记录

版本日期	变更说明
2023-12-20	第九次正式发布，本次变更如下： <ul style="list-style-type: none"><li>● 新增<b>列举资源记录器收集的全部资源</b>接口。</li><li>● 新增<b>查询资源记录器收集的资源数量</b>接口。</li><li>● 新增<b>列举资源记录器收集的资源标签</b>接口。</li><li>● 新增<b>列举资源记录器收集的资源概要</b>接口。</li><li>● 新增<b>查询资源记录器收集的单个资源</b>接口。</li><li>● “权限和授权项”下的“<b>资源清单</b>”章节新增资源记录器接口相关授权项。</li></ul>
2023-10-16	第八次正式发布，本次变更如下： <ul style="list-style-type: none"><li>● “<b>合规规则包</b>”下新增“组织合规规则包”相关接口。</li><li>● “权限和授权项”下的“<b>合规规则包</b>”章节新增组织合规规则包接口相关授权项。</li><li>● 更新“<b>API概览</b>”章节。</li></ul>
2023-08-04	第七次正式发布，本次变更如下： <ul style="list-style-type: none"><li>● 新增“<b>合规规则包</b>”相关接口。</li><li>● 新增“权限和授权项”下的“<b>合规规则包</b>”章节。</li><li>● 更新“<b>API概览</b>”章节。</li><li>● 更新“<b>错误码</b>”章节。</li></ul>
2023-06-07	第六次正式发布，本次变更如下： 服务名称变更：“资源管理服务 RMS”更名为“配置审计 Config”。

版本日期	变更说明
2023-02-10	第五次更新发布，本次更新如下： <ul style="list-style-type: none"><li>● 新增“<a href="#">查询账号下的单个资源</a>”接口。</li><li>● 新增“<a href="#">列举资源标签</a>”接口。</li><li>● 新增“<a href="#">查询资源数量</a>”接口。</li><li>● 新增“<a href="#">列举资源概要</a>”接口。</li><li>● 新增“<a href="#">列举资源关系详情</a>”接口。</li><li>● “<a href="#">合规性</a>”下新增“组织合规规则”相关接口。</li><li>● 新增“<a href="#">资源聚合器</a>”相关接口。</li><li>● 新增“<a href="#">权限和授权项</a>”下的“<a href="#">资源聚合器</a>”章节。</li></ul>
2022-12-07	第四次更新发布，本次更新如下： <ul style="list-style-type: none"><li>● 新增“<a href="#">更新合规评估结果</a>”接口。</li></ul>
2021-04-06	第三次正式发布，本次更新如下： <ul style="list-style-type: none"><li>● 新增“<a href="#">高级查询</a>”相关接口；</li><li>● 新增“<a href="#">权限和授权项</a>”下的“<a href="#">高级查询</a>”章节。</li></ul>
2020-12-25	第二次正式发布，本次更新如下： <ul style="list-style-type: none"><li>● 新增错误码：RMS.00010053；</li><li>● 新增“<a href="#">列举所有资源</a>”接口；</li><li>● 新增“<a href="#">区域管理</a>”下的“<a href="#">查询用户可见的区域</a>”接口。</li></ul>
2020-11-30	第一次正式发布。