

云连接

# 接口参考

文档版本 03  
发布日期 2024-03-06



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

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

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

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

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

# 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 ( Endpoint ) .....	1
1.4 约束与限制.....	1
1.5 基本概念.....	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 云连接实例.....	12
4.1.1 创建云连接实例.....	12
4.1.2 查询云连接列表.....	18
4.1.3 查询云连接实例.....	25
4.1.4 更新云连接实例.....	31
4.1.5 删除云连接实例.....	37
4.1.6 创建云连接实例标签.....	40
4.1.7 删除云连接实例标签.....	45
4.1.8 查询云连接实例的标签信息.....	49
4.1.9 通过标签过滤云连接实例.....	53
4.2 网络实例.....	60
4.2.1 创建网络实例.....	60
4.2.2 查询网络实例列表.....	67
4.2.3 查询网络实例.....	74
4.2.4 更新网络实例.....	79
4.2.5 删除网络实例.....	85
4.3 带宽包.....	88
4.3.1 创建带宽包实例.....	88
4.3.2 查询带宽包列表.....	98
4.3.3 查询带宽包实例.....	107

4.3.4 更新带宽包实例.....	113
4.3.5 删除带宽包实例.....	121
4.3.6 创建带宽包标签.....	124
4.3.7 删除带宽包标签.....	129
4.3.8 查询带宽包的标签信息.....	133
4.3.9 通过标签过滤带宽包实例.....	137
4.3.10 将带宽包实例绑定到云连接实例.....	145
4.3.11 解除带宽包实例与云连接实例的绑定.....	153
4.4 域间带宽.....	161
4.4.1 创建域间带宽实例.....	161
4.4.2 查询域间带宽列表.....	167
4.4.3 查询域间带宽实例.....	173
4.4.4 更新域间带宽实例.....	178
4.4.5 删除域间带宽实例.....	184
4.5 云连接路由.....	187
4.5.1 查询云连接路由条目列表.....	188
4.5.2 查询云连接路由条目详情.....	194
4.6 授权管理.....	198
4.6.1 创建授权.....	198
4.6.2 查询授权列表.....	204
4.6.3 查询权限列表.....	210
4.6.4 更新授权.....	217
4.6.5 删除授权.....	222
4.7 中心网络管理.....	225
4.7.1 创建中心网络.....	226
4.7.2 查询中心网络列表.....	236
4.7.3 查询中心网络详情.....	247
4.7.4 更新中心网络详情.....	255
4.7.5 删除中心网络.....	264
4.7.6 创建中心网络标签.....	272
4.7.7 删除中心网络标签.....	276
4.7.8 查询中心网络的标签信息.....	281
4.7.9 通过标签过滤中心网络实例.....	285
4.7.10 创建一个新版本的中心网络策略.....	295
4.7.11 查询所有版本的中心网络策略列表.....	304
4.7.12 应用中心网络策略.....	312
4.7.13 删除中心网络策略版本.....	320
4.7.14 查询中心网络策略变化集.....	324
4.8 中心网络的附件管理.....	329
4.8.1 创建中心网络 GDGW 附件.....	329
4.8.2 查询中心网络 GDGW 附件列表.....	337
4.8.3 创建中心网络 ER 路由表附件.....	345

4.8.4 查询中心网络 ER 路由表附件列表.....	354
4.8.5 查询中心网络 ER 路由表附件详情.....	362
4.8.6 更新中心网络 ER 路由表附件.....	368
4.8.7 查询中心网络 GDGW 附件详情.....	376
4.8.8 更新中心网络 GDGW 附件.....	382
4.8.9 查询中心网络附件列表.....	389
4.8.10 删除中心网络附件.....	398
4.9 中心网络连接.....	404
4.9.1 查询中心网络连接列表.....	404
4.9.2 更新中心网络连接接口.....	412
4.10 云连接配额.....	420
4.10.1 查询云连接配额.....	420
4.11 中心网络配额.....	426
4.11.1 查询中心网络配额.....	426
4.12 中心网络能力.....	429
4.12.1 查询中心网络能力列表.....	429
4.13 全域互联带宽.....	434
4.13.1 查询全域互联带宽列表.....	434
4.13.2 创建全域互联带宽.....	444
4.13.3 查询全域互联带宽详情.....	457
4.13.4 更新全域互联带宽详情.....	464
4.13.5 删除全域互联带宽.....	473
4.13.6 全域互联带宽绑定实例.....	476
4.13.7 全域互联带宽解绑实例.....	482
4.13.8 查询符合绑定条件的全域互联带宽列表.....	488
4.13.9 查询全域互联带宽租户配置信息.....	498
4.13.10 查询线路规格列表.....	503
4.13.11 查询站点列表.....	510
4.13.12 查询线路等级列表.....	517
4.14 全域互联带宽标签管理.....	523
4.14.1 查询账户全域互联带宽资源的标签.....	524
4.14.2 添加账户全域互联带宽资源标签.....	525
4.14.3 删除账户全域互联带宽资源标签.....	529
4.14.4 批量添加账户全域互联带宽资源标签.....	531
4.14.5 批量删除账户全域互联带宽资源标签.....	535
4.14.6 查询账户全域互联带宽资源标签数量.....	540
4.14.7 查询账户全域互联带宽资源实例列表.....	544
4.14.8 查询账户全域互联带宽所有资源标签.....	549
<b>5 权限与授权项.....</b>	<b>553</b>
5.1 权限与授权项说明.....	553
5.2 策略授权参考.....	554
<b>6 附录.....</b>	<b>564</b>

6.1 状态码.....	564
6.2 错误码.....	565
6.3 获取项目 ID.....	570
<b>7 历史 API.....</b>	<b>572</b>
7.1 标签管理.....	572
7.1.1 查询账户资源标签.....	572
7.1.2 删除资源标签.....	574
7.1.3 批量创建和删除资源标签.....	576
7.1.4 查询资源实例.....	583
7.1.5 查询资源标签.....	594
7.1.6 添加资源标签.....	596
7.2 全域互联带宽标签管理.....	600
7.2.1 查询账户全域互联带宽资源的标签.....	600
7.2.2 添加账户全域互联带宽资源标签.....	602
7.2.3 删除账户全域互联带宽资源标签.....	606
7.2.4 批量添加账户全域互联带宽资源标签.....	608
7.2.5 批量删除账户全域互联带宽资源标签.....	612
7.2.6 查询账户全域互联带宽资源标签数量.....	617
7.2.7 查询账户全域互联带宽资源实例列表.....	621
7.2.8 查询账户全域互联带宽所有资源标签.....	626
<b>A 修订记录.....</b>	<b>630</b>

# 1 使用前必读

## 1.1 概述

欢迎使用云连接（Cloud Connect）服务。云连接为用户提供一种能够快速构建跨区域VPC之间以及云上多VPC与云下多数据中心之间的高速、优质、稳定的网络能力，帮助用户打造一张具有企业级规模和通信能力的全球云上网络。

您可以使用本文档提供的API对云连接的资源进行相关操作，如创建、查询、修改、删除等。支持的全部操作请参见[2 API概览](#)。

在调用云连接服务的API之前，请确保已经充分了解云连接服务的相关概念，详细信息请参见[产品介绍](#)。

## 1.2 调用说明

云连接服务提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[3 如何调用API](#)。

同时云连接服务还提供多种编程语言的SDK供您使用，SDK的使用方法请参见<https://sdkcenter.developer.huaweicloud.com/?product=云连接>。

## 1.3 终端节点（Endpoint）

终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同，您可以从[地区和终端节点](#)中查询云连接的终端节点。

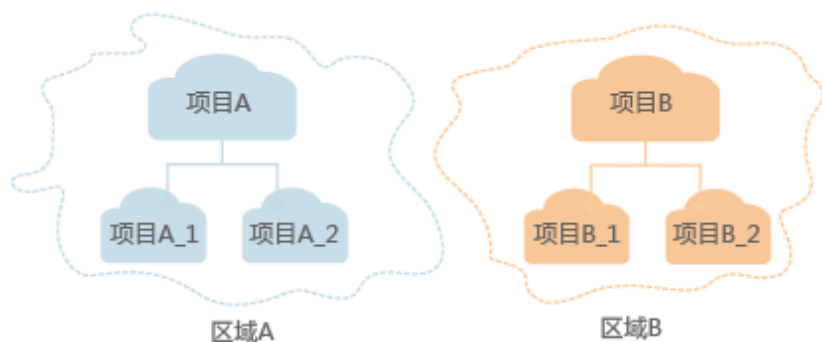
## 1.4 约束与限制

- 您能创建的云连接服务资源的数量与配额有关系，如果您想查看服务配额、扩大配额，具体请参见“[约束与限制](#)”。
- 更详细的限制请参见具体API的说明。

## 1.5 基本概念

- 账号  
用户注册时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用用户进行日常管理工作。
- 用户  
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。  
在[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](#)》。

# 2 API 概览

云连接提供自研的REST接口。

通过使用云连接所提供的接口，您可以完整的使用云连接的所有功能。云连接服务主要包括云连接实例、网络实例和云连接路由三种资源对象。

云连接提供的具体API如[表2-1](#)所示。

**表 2-1** 接口说明

子类型	说明
云连接实例接口	云连接实例接口，支持创建、查询、更新、删除云连接实例、查询列表等。
网络实例接口	网络实例接口，支持创建、查询、更新、删除网络实例、查询列表等。
云连接路由接口	云连接路由接口，支持查询路由条目详情、查询路由条目列表。

# 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。

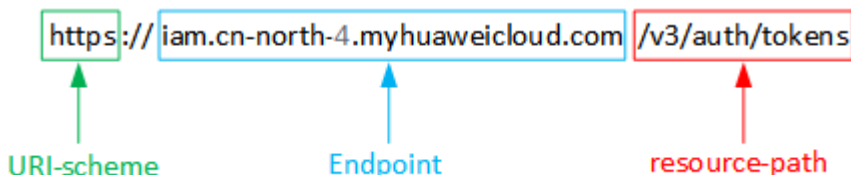
表 3-1 URI 中的参数说明

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。

调用本服务API需要项目级别的Token，即调用[获取用户Token](#)接口时，请求body中auth.scope的取值需要选择project，如下所示。

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username", //IAM用户名
          "password": SADMIN_PASS, //IAM用户密码，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" //IAM用户所属账号名
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx" //项目名称
      }
    }
  }
}
```

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 云连接实例

### 4.1.1 创建云连接实例

#### 功能介绍

创建云连接实例。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{domain\_id}/ccaas/cloud-connections

表 4-1 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>2000000</b>

表 4-3 请求 Body 参数

参数	是否必选	参数类型	描述
cloud_connection	是	CreateCloudConnection object	创建云连接实例的详细信息。

表 4-4 CreateCloudConnection

参数	是否必选	参数类型	描述
name	是	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度： <b>0</b> 最大长度： <b>255</b>
enterprise_project_id	否	String	实例所属企业项目ID。 最小长度： <b>1</b> 最大长度： <b>36</b>

## 响应参数

状态码： 201

表 4-5 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： <b>32</b> 最大长度： <b>36</b>
cloud_connection	CloudConnection object	云连接实例。

表 4-6 CloudConnection

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	云连接实例的状态。ACTIVE：表示状态可用。 最小长度：0 最大长度：36
admin_state_up	Boolean	云连接实例的管理状态。
used_scene	String	云连接使用场景。 <ul style="list-style-type: none"><li>• VPC：虚拟私有云。</li></ul> 缺省值： <b>vpc</b> 最小长度：0 最大长度：36
network_instance_number	Integer	云连接实例关联网络实例的个数。 最小值：0 最大值：10000
bandwidth_package_number	Integer	云连接实例关联带宽包的个数。 最小值：0 最大值：10000
inter_region_bandwidth_number	Integer	云连接实例关联域间带宽的个数。 最小值：0 最大值：10000

表 4-7 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

创建一个云连接实例

```
POST https://{cc_endpoint}/v3/{domain_id}/ccaas/cloud-connections
```

```
{
  "cloud_connection": {
    "name": "test",
    "description": "",
    "enterprise_project_id": "0"
  }
}
```

## 响应示例

状态码： 201

创建云连接实例成功。

```
{
  "cloud_connection": {
    "status": "ACTIVE",
    "id": "XXX",
    "name": "cloud-connection name",
    "description": "cloud-connection description",
    "domain_id": "XXX",
    "created_at": "2023-09-20T08:28:28Z",
    "updated_at": "2023-09-20T08:28:28Z",
    "admin_state_up": true,
    "used_scene": "vpc",
    "enterprise_project_id": "0",
    "network_instance_number": 0,
    "bandwidth_package_number": 0,
    "inter_region_bandwidth_number": 0,
    "tags": [ ]
  },
  "request_id": "d793a71f70e9370278b046ae39338393"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

### 创建一个云连接实例

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class CreateCloudConnectionSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateCloudConnectionRequest request = new CreateCloudConnectionRequest();
        CreateCloudConnectionRequestBody body = new CreateCloudConnectionRequestBody();
        CreateCloudConnection cloudConnectionbody = new CreateCloudConnection();
        cloudConnectionbody.setName("test")
            .withDescription("")
            .withEnterpriseProjectId("0");
        body.withCloudConnection(cloudConnectionbody);
        request.withBody(body);
        try {
            CreateCloudConnectionResponse response = client.createCloudConnection(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 创建一个云连接实例

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdccc.v3.region.cc_region import CcRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateCloudConnectionRequest()
        cloudConnectionbody = CreateCloudConnection(
            name="test",
            description="",
            enterprise_project_id="0"
        )
        request.body = CreateCloudConnectionRequestBody(
            cloud_connection=cloudConnectionbody
        )
        response = client.create_cloud_connection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 创建一个云连接实例

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.CreateCloudConnectionRequest{  
descriptionCloudConnection:= ""  
enterpriseProjectIdCloudConnection:= "0"  
cloudConnectionbody := &model.CreateCloudConnection{  
    Name: "test",  
    Description: &descriptionCloudConnection,  
    EnterpriseProjectId: &enterpriseProjectIdCloudConnection,  
}  
request.Body = &model.CreateCloudConnectionRequestBody{  
    CloudConnection: cloudConnectionbody,  
}  
response, err := client.CreateCloudConnection(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

## 更多

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

## 状态码

状态码	描述
201	创建云连接实例成功。

## 错误码

请参见[错误码](#)。

## 4.1.2 查询云连接列表

### 功能介绍

查询云连接列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/cloud-connections



表 4-8 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-9 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20
description	否	Array	根据描述查询，可查询多个描述。 数组长度：1 - 20
enterprise_project_id	否	Array	根据企业项目ID过滤列表。 数组长度：1 - 20
status	否	Array	根据状态过滤云连接实例列表。 ACTIVE：表示状态可用。 数组长度：1 - 10
type	否	Array	根据类型过滤云连接实例列表。 数组长度：1 - 10

参数	是否必选	参数类型	描述
used_scene	否	Array	根据使用场景过滤云连接实例列表。 数组长度：1 - 2

## 请求参数

表 4-10 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-11 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	<b>PageInfo</b> object	分页查询页的信息。
cloud_connections	Array of <b>CloudConnection</b> objects	云连接实例列表。 数组长度：0 - 2000

表 4-12 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096

参数	参数类型	描述
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-13 CloudConnection

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	云连接实例的状态。ACTIVE：表示状态可用。 最小长度：0 最大长度：36
admin_state_up	Boolean	云连接实例的管理状态。
used_scene	String	云连接使用场景。 <ul style="list-style-type: none"><li>● VPC：虚拟私有云。</li></ul> 缺省值：vpc 最小长度：0 最大长度：36

参数	参数类型	描述
network_instance_number	Integer	云连接实例关联网络实例的个数。 最小值： <b>0</b> 最大值： <b>10000</b>
bandwidth_package_number	Integer	云连接实例关联带宽包的个数。 最小值： <b>0</b> 最大值： <b>10000</b>
inter_region_bandwidth_number	Integer	云连接实例关联域间带宽的个数。 最小值： <b>0</b> 最大值： <b>10000</b>

表 4-14 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

### 查询云连接实例列表

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/cloud-connections
```

## 响应示例

### 状态码：200

查询云连接实例列表成功。

```
{
  "request_id": "ab44eb7776995d84a7580fe6a7bd7629",
  "cloud_connections": [ {
    "status": "ACTIVE",
    "id": "XXX",
    "name": "cloud-connenction name",
    "description": "cloud-connection description",
    "domain_id": "XXX",
    "created_at": "2023-09-20T08:28:28Z",
    "updated_at": "2023-09-20T08:28:28Z",
    "admin_state_up": true,
    "used_scene": "vpc",
    "enterprise_project_id": "0",
    "network_instance_number": 0,
    "bandwidth_package_number": 0,
    "inter_region_bandwidth_number": 0,
  }
]
```

```
"tags" : [ ]
}],
"page_info" : {
  "previous_marker" : "XXX",
  "current_count" : 30
}
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListCloudConnectionsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCloudConnectionsRequest request = new ListCloudConnectionsRequest();
        request.withStatus();
        request.withType();
        request.withUsedScene();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withId();
        request.withName();
        request.withDescription();
        request.withEnterpriseProjectId();
        try {
            ListCloudConnectionsResponse response = client.listCloudConnections(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCloudConnectionsRequest()
        request.status =
        request.type =
        request.used_scene =
        request.limit = <limit>
        request.marker = "<marker>"
        request.id =
        request.name =
        request.description =
        request.enterprise_project_id =
        response = client.list_cloud_connections(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询云连接实例列表成功。

## 错误码

请参见[错误码](#)。

### 4.1.3 查询云连接实例

#### 功能介绍

查询云连接实例。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{domain\_id}/ccaas/cloud-connections/{id}

表 4-15 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-16 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-17 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
cloud_connection	CloudConnection object	云连接实例。

表 4-18 CloudConnection

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。



参数	参数类型	描述
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	云连接实例的状态。ACTIVE：表示状态可用。 最小长度：0 最大长度：36
admin_state_up	Boolean	云连接实例的管理状态。
used_scene	String	云连接使用场景。 <ul style="list-style-type: none"><li>• VPC：虚拟私有云。</li></ul> 缺省值： <b>vpc</b> 最小长度：0 最大长度：36
network_instance_number	Integer	云连接实例关联网络实例的个数。 最小值：0 最大值：10000
bandwidth_package_number	Integer	云连接实例关联带宽包的个数。 最小值：0 最大值：10000
inter_region_bandwidth_number	Integer	云连接实例关联域间带宽的个数。 最小值：0 最大值：10000

表 4-19 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

查询云连接实例详情

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/cloud-connections/{id}
```

## 响应示例

状态码： 200

查询云连接实例详情成功。

```
{
  "cloud_connection": {
    "status": "ACTIVE",
    "id": "XXX",
    "name": "cloud-connection-name",
    "description": "cloud-connection-description",
    "domain_id": "XXX",
    "created_at": "2023-09-20T08:28:28Z",
    "updated_at": "2023-09-20T08:28:28Z",
    "admin_state_up": true,
    "used_scene": "vpc",
    "enterprise_project_id": "0",
    "network_instance_number": 0,
    "bandwidth_package_number": 0,
    "inter_region_bandwidth_number": 0,
    "tags": []
  },
  "request_id": "c4a34234aa46915cb04531216c1cc379"
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;
```

```
public class ShowCloudConnectionSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowCloudConnectionRequest request = new ShowCloudConnectionRequest();
        try {
            ShowCloudConnectionResponse response = client.showCloudConnection(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCloudConnectionRequest()
        response = client.show_cloud_connection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询云连接实例详情成功。

## 错误码

请参见[错误码](#)。

## 4.1.4 更新云连接实例

### 功能介绍

更新云连接实例。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{domain\_id}/ccaas/cloud-connections/{id}

表 4-20 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

### 请求参数

表 4-21 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-22 请求 Body 参数

参数	是否必选	参数类型	描述
cloud_connection	是	<b>UpdateCloud Connection</b> object	更新云连接实例的详细信息。

表 4-23 UpdateCloudConnection

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255

## 响应参数

状态码：200

表 4-24 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
cloud_connection	CloudConnection object	云连接实例。

表 4-25 CloudConnection

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。

参数	参数类型	描述
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	云连接实例的状态。ACTIVE：表示状态可用。 最小长度：0 最大长度：36
admin_state_up	Boolean	云连接实例的管理状态。
used_scene	String	云连接使用场景。 <ul style="list-style-type: none"><li>VPC：虚拟私有云。</li></ul> 缺省值：vpc 最小长度：0 最大长度：36
network_instance_number	Integer	云连接实例关联网络实例的个数。 最小值：0 最大值：10000
bandwidth_package_number	Integer	云连接实例关联带宽包的个数。 最小值：0 最大值：10000
inter_region_bandwidth_number	Integer	云连接实例关联域间带宽的个数。 最小值：0 最大值：10000

表 4-26 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

更新云连接实例的名称和描述

```
PUT https://{cc_endpoint}/v3/{domain_id}/ccaas/cloud-connections/{id}
{
```

```
"cloud_connection" : {  
  "name" : "new cloud-connection name",  
  "description" : "new cloud-connection description"  
}
```

## 响应示例

**状态码： 200**

更新云连接实例成功。

```
{  
  "cloud_connection" : {  
    "status" : "ACTIVE",  
    "id" : "XXX",  
    "name" : "new cloud-connection name",  
    "description" : "new cloud-connection description",  
    "domain_id" : "XXX",  
    "created_at" : "2023-09-20T08:28:28Z",  
    "updated_at" : "2023-09-20T09:24:38Z",  
    "admin_state_up" : true,  
    "used_scene" : "vpc",  
    "enterprise_project_id" : "0",  
    "network_instance_number" : 0,  
    "bandwidth_package_number" : 0,  
    "inter_region_bandwidth_number" : 0,  
    "tags" : []  
  },  
  "request_id" : "0e44ca4e6c250521d154e96a7e2b3cf0"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新云连接实例的名称和描述

```
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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
public class UpdateCloudConnectionSolution {  
  
  public static void main(String[] args) {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    // environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running  
    // this example, set environment variables CLOUD_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);
```



```
CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateCloudConnectionRequest request = new UpdateCloudConnectionRequest();
UpdateCloudConnectionRequestBody body = new UpdateCloudConnectionRequestBody();
UpdateCloudConnection cloudConnectionbody = new UpdateCloudConnection();
cloudConnectionbody.setName("new cloud-connection name")
    .withDescription("new cloud-connection description");
body.withCloudConnection(cloudConnectionbody);
request.withBody(body);
try {
    UpdateCloudConnectionResponse response = client.updateCloudConnection(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 更新云连接实例的名称和描述

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateCloudConnectionRequest()
        cloudConnectionbody = UpdateCloudConnection(
            name="new cloud-connection name",
            description="new cloud-connection description"
        )
        request.body = UpdateCloudConnectionRequestBody(
            cloud_connection=cloudConnectionbody
        )
        response = client.update_cloud_connection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

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

## Go

### 更新云连接实例的名称和描述

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateCloudConnectionRequest{}
    nameCloudConnection := "new cloud-connection name"
    descriptionCloudConnection := "new cloud-connection description"
    cloudConnectionbody := &model.UpdateCloudConnection{
        Name: &nameCloudConnection,
        Description: &descriptionCloudConnection,
    }
    request.Body = &model.UpdateCloudConnectionRequestBody{
        CloudConnection: cloudConnectionbody,
    }
    response, err := client.UpdateCloudConnection(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	更新云连接实例成功。

## 错误码

请参见[错误码](#)。

## 4.1.5 删除云连接实例

### 功能介绍

删除云连接实例。

### 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/{domain\_id}/ccaas/cloud-connections/{id}

表 4-27 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-28 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

无

## 请求示例

删除云连接实例

```
DELETE https://{cc_endpoint}/v3/{domain_id}/ccaas/cloud-connections/{id}
```

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteCloudConnectionSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteCloudConnectionRequest request = new DeleteCloudConnectionRequest();
        try {
            DeleteCloudConnectionResponse response = client.deleteCloudConnection(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcc.v3 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CcRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = DeleteCloudConnectionRequest()  
        response = client.delete_cloud_connection(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.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"  
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment 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 := cc.NewCcClient(  
        cc.CcClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).
```

```
Build()

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

## 更多

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

## 状态码

状态码	描述
204	删除云连接实例成功。

## 错误码

请参见[错误码](#)。

## 4.1.6 创建云连接实例标签

### 功能介绍

创建云连接实例标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/cloud-connections/{id}/tag

表 4-29 路径参数

参数	是否必选	参数类型	描述
id	是	String	资源的Id。 最小长度：32 最大长度：36
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-30 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>2000000</b>

表 4-31 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>Tag</b> objects	包含标签。 数组长度： <b>1 - 20</b>

表 4-32 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 响应参数

状态码： 204

表 4-33 响应 Header 参数

参数	参数类型	描述
x-request-id	-	-

## 请求示例

创建云连接实例标签

```
POST /v3/{domain_id}/ccaas/cloud-connections/{id}/tag
```

```
{
  "tags" : [ {
    "key" : "key",
    "value" : "value"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

#### 创建云连接实例标签

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class TagCloudConnectionSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        TagCloudConnectionRequest request = new TagCloudConnectionRequest();
        TagCloudConnectionRequestBody body = new TagCloudConnectionRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key")
                .withValue("value")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            TagCloudConnectionResponse response = client.tagCloudConnection(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```



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

## Python

### 创建云连接实例标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = TagCloudConnectionRequest()
        listTagsbody = [
            Tag(
                key="key",
                value="value"
            )
        ]
        request.body = TagCloudConnectionRequestBody(
            tags=listTagsbody
        )
        response = client.tag_cloud_connection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 创建云连接实例标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
```

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.TagCloudConnectionRequest{}
    valueTags := "value"
    var listTagsbody = []model.Tag{
        {
            Key: "key",
            Value: &valueTags,
        },
    }
    request.Body = &model.TagCloudConnectionRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.TagCloudConnection(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
204	创建云连接实例标签成功。

## 错误码

请参见[错误码](#)。

## 4.1.7 删除云连接实例标签

### 功能介绍

删除云连接实例标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/cloud-connections/{id}/untag

表 4-34 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

### 请求参数

表 4-35 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-36 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>Tag</b> objects	包含标签。 数组长度：1 - 20

表 4-37 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 响应参数

状态码： 204

表 4-38 响应 Header 参数

参数	参数类型	描述
x-request-id	-	-

## 请求示例

删除云连接实例标签

```
POST /v3/{domain_id}/ccaas/cloud-connections/{id}/untag
{
  "tags": [ {
    "key": "key",
    "value": "value"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

删除云连接实例标签

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class UntagCloudConnectionSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        UntagCloudConnectionRequest request = new UntagCloudConnectionRequest();
        UntagCloudConnectionRequestBody body = new UntagCloudConnectionRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key")
                .withValue("value")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            UntagCloudConnectionResponse response = client.untagCloudConnection(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 删除云连接实例标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

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

```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UntagCloudConnectionRequest()
    listTagsbody = [
        Tag(
            key="key",
            value="value"
        )
    ]
    request.body = UntagCloudConnectionRequestBody(
        tags=listTagsbody
    )
    response = client.untag_cloud_connection(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 删除云连接实例标签

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UntagCloudConnectionRequest{
        valueTags:= "value"
        var listTagsbody = []model.Tag{
```

```
{
  Key: "key",
  Value: &valueTags,
},
}
request.Body = &model.UntagCloudConnectionRequestBody{
  Tags: listTagsbody,
}
response, err := client.UntagCloudConnection(request)
if err == nil {
  fmt.Printf("%+v\n", response)
} else {
  fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
204	删除云连接实例标签成功。

## 错误码

请参见[错误码](#)。

## 4.1.8 查询云连接实例的标签信息

### 功能介绍

查询云连接实例的标签信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/cloud-connections/tags

表 4-39 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-40 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-41 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
tags	Array of <b>MultivaluedTag</b> objects	云连接实例的所有标签。 数组长度：1 - 200

表 4-42 MultivaluedTag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
values	Array of strings	相同键的值列表 数组长度：1 - 20

## 请求示例

查询云连接实例标签

```
GET /v3/{domain_id}/ccaas/cloud-connections/tags
```

## 响应示例

状态码：200

查询云连接实例标签信息成功。



```
{
  "request_id" : "2f68486e542a0bb23ab86a826f909ecf",
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1" ]
  }, {
    "key" : "key2",
    "values" : [ "value2" ]
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ListCloudConnectionTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCloudConnectionTagsRequest request = new ListCloudConnectionTagsRequest();
        try {
            ListCloudConnectionTagsResponse response = client.listCloudConnectionTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCloudConnectionTagsRequest()
        response = client.list_cloud_connection_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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCloudConnectionTagsRequest{}
    response, err := client.ListCloudConnectionTags(request)
```

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

## 更多

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

## 状态码

状态码	描述
200	查询云连接实例标签信息成功。

## 错误码

请参见[错误码](#)。

## 4.1.9 通过标签过滤云连接实例

### 功能介绍

通过标签过滤云连接实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/cloud-connections/filter

表 4-43 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-44 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-45 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>MultivaluedTag</b> objects	包含标签。 数组长度：1 - 20

表 4-46 MultivaluedTag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
values	是	Array of strings	相同键的值列表 数组长度：1 - 20

## 响应参数

状态码：200

表 4-47 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	<b>PageInfo</b> object	分页查询页的信息。

参数	参数类型	描述
cloud_connections	Array of <b>CloudConnection</b> objects	云连接实例列表。 数组长度：0 - 2000

表 4-48 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-49 CloudConnection

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20

参数	参数类型	描述
status	String	云连接实例的状态。ACTIVE：表示状态可用。 最小长度：0 最大长度：36
admin_state_up	Boolean	云连接实例的管理状态。
used_scene	String	云连接使用场景。 <ul style="list-style-type: none"><li>VPC：虚拟私有云。</li></ul> 缺省值：vpc 最小长度：0 最大长度：36
network_instance_number	Integer	云连接实例关联网络实例的个数。 最小值：0 最大值：10000
bandwidth_package_number	Integer	云连接实例关联带宽包的个数。 最小值：0 最大值：10000
inter_region_bandwidth_number	Integer	云连接实例关联域间带宽的个数。 最小值：0 最大值：10000

表 4-50 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

通过标签过滤云连接实例

```
POST https://{cc_endpoint}/v3/{domain_id}/ccaas/cloud-connections/filter
```

```
{  
  "tags": [{  
    "key": "key",  
    "values": [ "value" ]  
  }  
}
```

```
    }  
  }  
}
```

## 响应示例

**状态码： 200**

通过标签过滤云连接实例成功。

```
{  
  "request_id": "ab44eb7776995d84a7580fe6a7bd7629",  
  "cloud_connections": [ {  
    "status": "ACTIVE",  
    "id": "XXX",  
    "name": "cloud-connenction name",  
    "description": "cloud-connection description",  
    "domain_id": "XXX",  
    "created_at": "2023-09-20T08:28:28Z",  
    "updated_at": "2023-09-20T08:28:28Z",  
    "admin_state_up": true,  
    "used_scene": "vpc",  
    "enterprise_project_id": "0",  
    "network_instance_number": 0,  
    "bandwidth_package_number": 0,  
    "inter_region_bandwidth_number": 0,  
    "tags": [ {  
      "key": "key1",  
      "value": "value1"  
    }, {  
      "key": "key2",  
      "value": "value2"  
    } ]  
  } ],  
  "page_info": {  
    "previous_marker": "XXX",  
    "current_count": 30  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

通过标签过滤云连接实例

```
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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ListCloudConnectionsByTagsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListCloudConnectionsByTagsRequest request = new ListCloudConnectionsByTagsRequest();
ListCloudConnectionsByTagsRequestBody body = new ListCloudConnectionsByTagsRequestBody();
List<String> listTagsValues = new ArrayList<>();
listTagsValues.add("value");
List<MultivaluedTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new MultivaluedTag()
        .withKey("key")
        .withValues(listTagsValues)
);
body.withTags(listbodyTags);
request.withBody(body);
try {
    ListCloudConnectionsByTagsResponse response = client.listCloudConnectionsByTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 通过标签过滤云连接实例

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()
```



```
try:
    request = ListCloudConnectionsByTagsRequest()
    listValuesTags = [
        "value"
    ]
    listTagsbody = [
        MultivaluedTag(
            key="key",
            values=listValuesTags
        )
    ]
    request.body = ListCloudConnectionsByTagsRequestBody(
        tags=listTagsbody
    )
    response = client.list_cloud_connections_by_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

### 通过标签过滤云连接实例

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCloudConnectionsByTagsRequest{}
    var listValuesTags = []string{
        "value",
    }
    var listTagsbody = []model.MultivaluedTag{
        {
            Key: "key",
            Values: listValuesTags,
        },
    }
    request.Body = &model.ListCloudConnectionsByTagsRequestBody{
        Tags: listTagsbody,
```

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

## 更多

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

## 状态码

状态码	描述
200	通过标签过滤云连接实例成功。

## 错误码

请参见[错误码](#)。

# 4.2 网络实例

## 4.2.1 创建网络实例

### 功能介绍

创建网络实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/network-instances

表 4-51 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-52 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>2000000</b>

表 4-53 请求 Body 参数

参数	是否必选	参数类型	描述
network_instance	是	CreateNetworkInstance object	创建网络实例的详细信息。

表 4-54 CreateNetworkInstance

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度： <b>0</b> 最大长度： <b>255</b>
instance_id	是	String	资源ID标识符。
instance_domain_id	否	String	实例所属账号ID。
project_id	是	String	实例所属项目ID。
region_id	是	String	RegionID。
cloud_connection_id	是	String	资源ID标识符。
type	是	String	添加到云连接网络实例的类型，有效值： <ul style="list-style-type: none"><li>vpc：虚拟私有云。</li><li>vgw：虚拟网关。</li></ul> 最小长度： <b>0</b> 最大长度： <b>36</b>

参数	是否必选	参数类型	描述
cidrs	是	Array of strings	网络实例发布的网段路由列表。 最小长度：0 最大长度：64 数组长度：0 - 1000

## 响应参数

状态码：201

表 4-55 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
network_instance	<a href="#">NetworkInstance</a> object	网络实例。

表 4-56 NetworkInstance

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
cloud_connection_id	String	资源ID标识符。
instance_id	String	资源ID标识符。
instance_domain_id	String	实例所属账号ID。

参数	参数类型	描述
region_id	String	RegionID。
project_id	String	实例所属项目ID。
status	String	网络实例的状态。 <ul style="list-style-type: none"><li>● ACTIVE: 处理成功。</li><li>● PENDING: 处理中。</li><li>● ERROR: 处理失败。</li></ul> 最小长度: 0 最大长度: 36
type	String	网络实例的类型。 <ul style="list-style-type: none"><li>● vpc: 虚拟私有云。</li><li>● vgw: 虚拟网关。</li></ul> 最小长度: 0 最大长度: 36
cidrs	Array of strings	网络实例发布的网段路由列表。 最小长度: 0 最大长度: 64 数组长度: 0 - 1000

## 请求示例

创建一个虚拟私有云实例，并指定该实例发布的网段路由列表

POST https://{cc\_endpoint}/v3/{domain\_id}/ccaas/network-instances

```
{
  "network_instance" : {
    "type" : "vpc",
    "region_id" : "cn-north-4",
    "project_id" : "XXX",
    "cloud_connection_id" : "XXX",
    "instance_id" : "XXX",
    "cidrs" : [ "192.168.1.0/24" ]
  }
}
```

## 响应示例

**状态码: 201**

创建网络实例成功。

```
{
  "request_id" : "c8fe3d71352e273915bfbd74ce58368a",
  "network_instance" : {
    "id" : "XXX",
    "name" : "",
    "description" : "",
    "domain_id" : "XXX",
  }
}
```

```
"created_at" : "2023-09-20T09:33:15Z",
"updated_at" : "2023-09-20T09:33:16Z",
"project_id" : "XXX",
"instance_domain_id" : "",
"cloud_connection_id" : "XXX",
"region_id" : "cn-north-4",
"type" : "vpc",
"instance_id" : "XXX",
"cidrs" : [ "192.168.1.0/24" ],
"status" : "ACTIVE"
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个虚拟私有云实例，并指定该实例发布的网段路由列表

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class CreateNetworkInstanceSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateNetworkInstanceRequest request = new CreateNetworkInstanceRequest();
        CreateNetworkInstanceRequestBody body = new CreateNetworkInstanceRequestBody();
        List<String> listNetworkInstanceCidrs = new ArrayList<>();
        listNetworkInstanceCidrs.add("192.168.1.0/24");
        CreateNetworkInstance networkInstancebody = new CreateNetworkInstance();
        networkInstancebody.withInstanceId("XXX")
            .withProjectId("XXX")
            .withRegionId("cn-north-4")
            .withCidrs(listNetworkInstanceCidrs)
            .withType(CreateNetworkInstance.TypeEnum.fromValue("vpc"))
            .withCloudConnectionId("XXX");
        body.withNetworkInstance(networkInstancebody);
        request.withBody(body);
    }
}
```

```
try {
    CreateNetworkInstanceResponse response = client.createNetworkInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

创建一个虚拟私有云实例，并指定该实例发布的网段路由列表

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateNetworkInstanceRequest()
        listCidrsNetworkInstance = [
            "192.168.1.0/24"
        ]
        networkInstancebody = CreateNetworkInstance(
            instance_id="XXX",
            project_id="XXX",
            region_id="cn-north-4",
            cidrs=listCidrsNetworkInstance,
            type="vpc",
            cloud_connection_id="XXX"
        )
        request.body = CreateNetworkInstanceRequestBody(
            network_instance=networkInstancebody
        )
        response = client.create_network_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

创建一个虚拟私有云实例，并指定该实例发布的网段路由列表

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNetworkInstanceRequest{}
    var listCidrsNetworkInstance = []string{
        "192.168.1.0/24",
    }
    networkInstancebody := &model.CreateNetworkInstance{
        InstanceId: "XXX",
        ProjectId: "XXX",
        RegionId: "cn-north-4",
        Cidrs: listCidrsNetworkInstance,
        Type: model.GetCreateNetworkInstanceTypeEnum().VPC,
        CloudConnectionId: "XXX",
    }
    request.Body = &model.CreateNetworkInstanceRequestBody{
        NetworkInstance: networkInstancebody,
    }
    response, err := client.CreateNetworkInstance(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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



## 状态码

状态码	描述
201	创建网络实例成功。

## 错误码

请参见[错误码](#)。

## 4.2.2 查询网络实例列表

### 功能介绍

查询云连接列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/network-instances

表 4-57 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-58 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000

参数	是否必选	参数类型	描述
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20
description	否	Array	根据描述查询，可查询多个描述。 数组长度：1 - 20
cloud_connection_id	否	Array	根据云连接的ID过滤列表。 数组长度：1 - 20
status	否	Array	根据状态过滤网络实例列表。 ACTIVE：表示状态可用。 数组长度：1 - 10
type	否	Array	根据类型过滤网络实例列表。 数组长度：1 - 10
instance_id	否	Array	根据网络实例ID过滤网络实例列表。 数组长度：1 - 10
region_id	否	Array	根据网络实例所在的Region过滤网络实例列表。 数组长度：1 - 10

## 请求参数

表 4-59 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-60 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
network_instances	Array of NetworkInstance objects	网络实例列表。 数组长度：0 - 2000

表 4-61 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-62 NetworkInstance

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
cloud_connection_id	String	资源ID标识符。
instance_id	String	资源ID标识符。
instance_domain_id	String	实例所属账号ID。
region_id	String	RegionID。
project_id	String	实例所属项目ID。
status	String	网络实例的状态。 <ul style="list-style-type: none"><li>● ACTIVE：处理成功。</li><li>● PENDING：处理中。</li><li>● ERROR：处理失败。</li></ul> 最小长度：0 最大长度：36
type	String	网络实例的类型。 <ul style="list-style-type: none"><li>● vpc：虚拟私有云。</li><li>● vgw：虚拟网关。</li></ul> 最小长度：0 最大长度：36
cidrs	Array of strings	网络实例发布的网段路由列表。 最小长度：0 最大长度：64 数组长度：0 - 1000

## 请求示例

### 查询网络实例列表

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/network-instances
```

## 响应示例

### 状态码： 200

查询网络实例列表成功。

```
{
  "request_id" : "b961d5fcda6b993b5b750f1b96e484e0",
  "network_instances" : [ {
    "id" : "XXX",
    "name" : "",
    "description" : "",
    "domain_id" : "XXX",
    "created_at" : "2023-09-20T09:33:15Z",
    "updated_at" : "2023-09-20T09:33:16Z",
    "project_id" : "XXX",
    "instance_domain_id" : "",
    "cloud_connection_id" : "XXX",
    "region_id" : "cn-north-4",
    "type" : "vpc",
    "instance_id" : "XXX",
    "cidrs" : [ "192.168.1.0/24" ],
    "status" : "ACTIVE"
  } ],
  "page_info" : {
    "previous_marker" : "XXX",
    "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListNetworkInstancesSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListNetworkInstancesRequest request = new ListNetworkInstancesRequest();
request.withStatus();
request.withType();
request.withInstanceId();
request.withRegionId();
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withId();
request.withName();
request.withDescription();
request.withCloudConnectionId();
try {
    ListNetworkInstancesResponse response = client.listNetworkInstances(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListNetworkInstancesRequest()
        request.status =
        request.type =
        request.instance_id =
        request.region_id =
        request.limit = <limit>
```

```
request.marker = "<marker>"
request.id =
request.name =
request.description =
request.cloud_connection_id =
response = client.list_network_instances(request)
print(response)
except exceptions.ClientRequestException as e:
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询网络实例列表成功。

## 错误码

请参见[错误码](#)。

## 4.2.3 查询网络实例

### 功能介绍

查询网络实例。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/ccaas/network-instances/{id}

表 4-63 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-64 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000



## 响应参数

状态码： 200

表 4-65 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
network_instance	<b>NetworkInstance</b> object	网络实例。

表 4-66 NetworkInstance

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
domain_id	String	实例所属账号ID。
created_at	String	实例创建时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
cloud_connection_id	String	资源ID标识符。
instance_id	String	资源ID标识符。
instance_domain_id	String	实例所属账号ID。
region_id	String	RegionID。
project_id	String	实例所属项目ID。

参数	参数类型	描述
status	String	网络实例的状态。 <ul style="list-style-type: none"><li>● ACTIVE: 处理成功。</li><li>● PENDING: 处理中。</li><li>● ERROR: 处理失败。</li></ul> 最小长度: 0 最大长度: 36
type	String	网络实例的类型。 <ul style="list-style-type: none"><li>● vpc: 虚拟私有云。</li><li>● vgw: 虚拟网关。</li></ul> 最小长度: 0 最大长度: 36
cidrs	Array of strings	网络实例发布的网段路由列表。 最小长度: 0 最大长度: 64 数组长度: 0 - 1000

## 请求示例

查询网络实例详情

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/network-instances/{id}
```

## 响应示例

状态码: 200

查询网络实例详情成功。

```
{
  "request_id": "b961d5fcda6b993b5b750f1b96e484e0",
  "network_instance": {
    "id": "XXX",
    "name": "",
    "description": "",
    "domain_id": "XXX",
    "created_at": "2023-09-20T09:33:15Z",
    "updated_at": "2023-09-20T09:33:16Z",
    "project_id": "XXX",
    "instance_domain_id": "",
    "cloud_connection_id": "XXX",
    "region_id": "cn-north-4",
    "type": "vpc",
    "instance_id": "XXX",
    "cidrs": [ "192.168.1.0/24" ],
    "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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ShowNetworkInstanceSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowNetworkInstanceRequest request = new ShowNetworkInstanceRequest();
        try {
            ShowNetworkInstanceResponse response = client.showNetworkInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

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

```
# In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

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

## Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询网络实例详情成功。

## 错误码

请参见[错误码](#)。

## 4.2.4 更新网络实例

### 功能介绍

更新网络实例。

### 调用方法

请参见[如何调用API](#)。

## URI

PUT /v3/{domain\_id}/ccaas/network-instances/{id}

表 4-67 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-68 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-69 请求 Body 参数

参数	是否必选	参数类型	描述
network_instance	是	<b>UpdateNetworkInstance</b> object	更新网络实例的详细信息。

表 4-70 UpdateNetworkInstance

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
cidrs	否	Array of strings	网络实例发布的网段路由列表。 最小长度：0 最大长度：64 数组长度：0 - 1000

## 响应参数

状态码：200

表 4-71 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
network_instance	<b>NetworkInstance</b> object	网络实例。

表 4-72 NetworkInstance

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。

参数	参数类型	描述
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
cloud_connection_id	String	资源ID标识符。
instance_id	String	资源ID标识符。
instance_domain_id	String	实例所属账号ID。
region_id	String	RegionID。
project_id	String	实例所属项目ID。
status	String	网络实例的状态。 <ul style="list-style-type: none"><li>● ACTIVE：处理成功。</li><li>● PENDING：处理中。</li><li>● ERROR：处理失败。</li></ul> 最小长度：0 最大长度：36
type	String	网络实例的类型。 <ul style="list-style-type: none"><li>● vpc：虚拟私有云。</li><li>● vgw：虚拟网关。</li></ul> 最小长度：0 最大长度：36
cidrs	Array of strings	网络实例发布的网段路由列表。 最小长度：0 最大长度：64 数组长度：0 - 1000

## 请求示例

更新网络实例的描述

```
PUT https://{cc_endpoint}/v3/{domain_id}/ccaas/network-instances/{id}
{
```

```
"network_instance" : {  
  "description" : "new description"  
}
```

## 响应示例

**状态码： 200**

更新网络实例成功。

```
{  
  "request_id" : "1e3cd137d7c27ad139886f23ade736b0",  
  "network_instance" : {  
    "id" : "XXX",  
    "name" : "",  
    "description" : "new description",  
    "domain_id" : "XXX",  
    "created_at" : "2023-09-20T09:33:15Z",  
    "updated_at" : "2023-09-21T03:28:08Z",  
    "project_id" : "XXX",  
    "instance_domain_id" : "",  
    "cloud_connection_id" : "XXX",  
    "region_id" : "cn-north-4",  
    "type" : "vpc",  
    "instance_id" : "0a6904ae-04d2-42ff-91d5-7d421c60a04c",  
    "cidrs" : [ "192.168.1.0/24" ],  
    "status" : "ACTIVE"  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新网络实例的描述

```
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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
public class UpdateNetworkInstanceSolution {  
  
  public static void main(String[] args) {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    // environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running  
    // this example, set environment variables CLOUD_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);
```



```
CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateNetworkInstanceRequest request = new UpdateNetworkInstanceRequest();
UpdateNetworkInstanceRequestBody body = new UpdateNetworkInstanceRequestBody();
UpdateNetworkInstance networkInstancebody = new UpdateNetworkInstance();
networkInstancebody.withDescription("new description");
body.withNetworkInstance(networkInstancebody);
request.withBody(body);
try {
    UpdateNetworkInstanceResponse response = client.updateNetworkInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 更新网络实例的描述

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.newBuilder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateNetworkInstanceRequest()
        networkInstancebody = UpdateNetworkInstance(
            description="new description"
        )
        request.body = UpdateNetworkInstanceRequestBody(
            network_instance=networkInstancebody
        )
        response = client.update_network_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 更新网络实例的描述

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNetworkInstanceRequest{
        descriptionNetworkInstance:= "new description"
    }
    networkInstancebody := &model.UpdateNetworkInstance{
        Description: &descriptionNetworkInstance,
    }
    request.Body = &model.UpdateNetworkInstanceRequestBody{
        NetworkInstance: networkInstancebody,
    }
    response, err := client.UpdateNetworkInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	更新网络实例成功。

## 错误码

请参见[错误码](#)。

## 4.2.5 删除网络实例

### 功能介绍

删除网络实例。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/ccaas/network-instances/{id}

表 4-73 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

### 请求参数

表 4-74 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

### 响应参数

无

### 请求示例

删除网络实例

```
DELETE https://{cc_endpoint}/v3/{domain_id}/ccaas/network-instances/{id}
```

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteNetworkInstanceSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteNetworkInstanceRequest request = new DeleteNetworkInstanceRequest();
        try {
            DeleteNetworkInstanceResponse response = client.deleteNetworkInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

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

## Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

```
}  
}
```

## 更多

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

## 状态码

状态码	描述
204	删除网络实例成功。

## 错误码

请参见[错误码](#)。

# 4.3 带宽包

## 4.3.1 创建带宽包实例

### 功能介绍

创建带宽包实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/bandwidth-packages

表 4-75 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-76 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>2000000</b>

表 4-77 请求 Body 参数

参数	是否必选	参数类型	描述
bandwidth_package	是	CreateBandwidthPackage object	创建带宽包请求体。

表 4-78 CreateBandwidthPackage

参数	是否必选	参数类型	描述
name	是	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度： <b>0</b> 最大长度： <b>255</b>
enterprise_project_id	否	String	实例所属企业项目ID。 最小长度： <b>1</b> 最大长度： <b>36</b>
project_id	是	String	实例所属项目ID。
local_area_id	是	String	本端大区。云连接支持的大区有：   Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北

参数	是否必选	参数类型	描述
remote_area_id	是	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
charge_mode	是	String	带宽包实例的计费方式。 bandwidth是按带宽计费。 最小长度： <b>0</b> 最大长度： <b>64</b>
billing_mode	是	Integer	带宽包实例在大陆站或国际站的计费方式： <ul style="list-style-type: none"><li>• 3：大陆站按需计费</li><li>• 4：国际站按需计费</li><li>• 5：大陆站按95方式计费</li><li>• 6：国际站按95方式计费</li></ul>
bandwidth	是	Integer	带宽包实例中的带宽值。 最小值： <b>2</b> 最大值： <b>999999</b>
resource_id	否	String	带宽包实例绑定的资源ID。 最小长度： <b>0</b> 最大长度： <b>36</b>
resource_type	否	String	带宽包实例绑定的资源类型。 cloud_connection: 云连接实例。 最小长度： <b>0</b> 最大长度： <b>36</b>



参数	是否必选	参数类型	描述
spec_code	否	String	带宽包实例的规格编码。 bandwidth.aftoela: 大陆站+国际站南非-拉美东 bandwidth.aftonla: 大陆站+国际站南非-拉美北 bandwidth.aftowla: 大陆站+国际站南非-拉美西 bandwidth.aptoaf: 国际站亚太-南非 bandwidth.aptoap: 国际站亚太-亚太 bandwidth.aptoela: 大陆站+国际站亚太-拉美东 bandwidth.aptonla: 大陆站+国际站亚太-拉美北 bandwidth.aptowla: 大陆站+国际站亚太-拉美西 bandwidth.cmtoaf: 国际站中国大陆-南非 bandwidth.cmtoap: 国际站中国大陆-亚太 bandwidth.cmtocm: 国际站中国大陆-中国大陆 bandwidth.cmtoela: 大陆站+国际站中国大陆-拉美东 bandwidth.cmtonla: 大陆站+国际站中国大陆-拉美北 bandwidth.cmtowla: 大陆站+国际站中国大陆-拉美西 bandwidth.elatoela: 大陆站+国际站拉美东-拉美东 bandwidth.elatonla: 大陆站+国际站拉美东-拉美北 bandwidth.wlatoela: 大陆站+国际站拉美西-拉美东 bandwidth.wlatonla: 大陆站+国际站拉美西-拉美北 bandwidth.wlatowla: 大陆站+国际站拉美西-拉美西 最小长度: 0 最大长度: 32
interflow_mode	否	String	互通类型: <ul style="list-style-type: none"> <li>• Area: 大区互通</li> <li>• Region: 城域互通</li> </ul> 最小长度: 0 最大长度: 36

## 响应参数

状态码： 201

表 4-79 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
bandwidth_package	<b>BandwidthPackage</b> object	带宽包实例。

表 4-80 BandwidthPackage

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度： 1 最大长度： 36
project_id	String	实例所属项目ID。
created_at	String	实例创建时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
resource_id	String	带宽包实例绑定的资源ID。 最小长度： 0 最大长度： 36
resource_type	String	带宽包实例绑定的资源类型。 cloud_connection: 云连接实例。 最小长度： 0 最大长度： 32

参数	参数类型	描述
local_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
remote_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
spec_code	String	带宽包实例的规格编码。bandwidth.aftoela：大陆站+国际站南非-拉美东 bandwidth.aftonla：大陆站+国际站南非-拉美北 bandwidth.aftowla：大陆站+国际站南非-拉美西 bandwidth.aptoaf：国际站亚太-南非 bandwidth.aptoap：国际站亚太-亚太 bandwidth.aptoela：大陆站+国际站亚太-拉美东 bandwidth.aptonla：大陆站+国际站亚太-拉美北 bandwidth.aptowla：大陆站+国际站亚太-拉美西 bandwidth.cmtoaf：国际站中国大陆-南非 bandwidth.cmtoap：国际站中国大陆-亚太 bandwidth.cmtocm：国际站中国大陆-中国大陆 bandwidth.cmtoela：大陆站+国际站中国大陆-拉美东 bandwidth.cmtonla：大陆站+国际站中国大陆-拉美北 bandwidth.cmtowla：大陆站+国际站中国大陆-拉美西 bandwidth.elatoela：大陆站+国际站拉美东-拉美东 bandwidth.elatonla：大陆站+国际站拉美东-拉美北 bandwidth.wlatoela：大陆站+国际站拉美西-拉美东 bandwidth.wlatonla：大陆站+国际站拉美西-拉美北 bandwidth.wlatowla：大陆站+国际站拉美西-拉美西 最小长度：0 最大长度：32
billing_mode	Integer	带宽包实例在大陆站或国际站的计费方式。1：大陆站包周期 2：国际站包周期 3：大陆站按需计费 4：国际站按需计费 5：大陆站按95方式计费 6：国际站按95方式计费
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	带宽包实例的状态。ACTIVE表示状态 最小长度：0 最大长度：36
admin_state_up	Boolean	带宽包实例的管理状态。

参数	参数类型	描述
order_id	String	带宽包实例的订单ID。 最小长度：0 最大长度：32
product_id	String	带宽包实例的产品ID。 最小长度：0 最大长度：36
charge_mode	String	带宽包实例的计费方式。bandwidth是按带宽计费。 最小长度：0 最大长度：64
bandwidth	Integer	带宽包实例中的带宽值。 最小值：2 最大值：999999
interflow_mode	String	互通类型： <ul style="list-style-type: none"><li>Area: 大区互通</li><li>Region: 城域互通</li></ul> 最小长度：0 最大长度：32

表 4-81 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

创建一个带宽包实例，带宽值为5M，并指定带宽包的本端大区 and 远端大区

```
POST https://{cc_endpoint}/v3/{domain_id}/ccaas/bandwidth-packages
```

```
{  
  "bandwidth_package": {  
    "project_id": "XXX",  
    "name": "test_bwp_name",  
    "local_area_id": "Chinese-Mainland",  
    "remote_area_id": "Chinese-Mainland",  
    "bandwidth": 5,  
  }  
}
```

```
"charge_mode": "bandwidth",  
"billing_mode": 5,  
"interflow_mode": "Area",  
"enterprise_project_id": "0"  
}  
}
```

## 响应示例

**状态码：201**

创建带宽包实例成功。

```
{  
  "request_id": "61126320a1802d5c6444f9d2d76526c2",  
  "bandwidth_package": {  
    "bandwidth": 5,  
    "status": "ACTIVE",  
    "id": "XXXXX",  
    "name": "test_bwp_name",  
    "description": "",  
    "domain_id": "XXX",  
    "created_at": "2023-09-20T08:37:15Z",  
    "updated_at": "2023-09-20T08:37:15Z",  
    "project_id": "XXX",  
    "resource_type": "",  
    "resource_id": "",  
    "local_area_id": "Chinese-Mainland",  
    "remote_area_id": "Chinese-Mainland",  
    "admin_state_up": true,  
    "charge_mode": "bandwidth",  
    "billing_mode": "3",  
    "spec_code": "bandwidth.cmtocm",  
    "order_id": "",  
    "product_id": "",  
    "enterprise_project_id": "0",  
    "tags": [],  
    "interflow_mode": "Area"  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个带宽包实例，带宽值为5M，并指定带宽包的本端大区 and 远端大区

```
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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
public class CreateBandwidthPackageSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
CreateBandwidthPackageRequest request = new CreateBandwidthPackageRequest();
CreateBandwidthPackageRequestBody body = new CreateBandwidthPackageRequestBody();
CreateBandwidthPackage bandwidthPackagebody = new CreateBandwidthPackage();
bandwidthPackagebody.withBillingMode(CreateBandwidthPackage.BillingModeEnum.NUMBER_5)
    .withBandwidth(5)
    .withChargeMode(CreateBandwidthPackage.ChargeModeEnum.fromValue("bandwidth"))
    .withRemoteAreaId(CreateBandwidthPackage.RemoteAreaIdEnum.fromValue("Chinese-Mainland"))
    .withEnterpriseProjectId("0")
    .withProjectId("XXX")
    .withName("test_bwp_name")
    .withLocalAreaId(CreateBandwidthPackage.LocalAreaIdEnum.fromValue("Chinese-Mainland"))
    .withInterflowMode(CreateBandwidthPackage.InterflowModeEnum.fromValue("Area"));
body.withBandwidthPackage(bandwidthPackagebody);
request.withBody(body);
try {
    CreateBandwidthPackageResponse response = client.createBandwidthPackage(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

创建一个带宽包实例，带宽值为5M，并指定带宽包的本端大区 and 远端大区

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

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

```
.with_region(CcRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = CreateBandwidthPackageRequest()
    bandwidthPackagebody = CreateBandwidthPackage(
        billing_mode=5,
        bandwidth=5,
        charge_mode="bandwidth",
        remote_area_id="Chinese-Mainland",
        enterprise_project_id="0",
        project_id="XXX",
        name="test_bwp_name",
        local_area_id="Chinese-Mainland",
        interflow_mode="Area"
    )
    request.body = CreateBandwidthPackageRequestBody(
        bandwidth_package=bandwidthPackagebody
    )
    response = client.create_bandwidth_package(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

创建一个带宽包实例，带宽值为5M，并指定带宽包的本端大区 and 远端大区

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateBandwidthPackageRequest{
        remoteAreaIdBandwidthPackage:=
model.GetCreateBandwidthPackageRemoteAreaIdEnum().CHINESE_MAINLAND
        enterpriseProjectIdBandwidthPackage:= "0"
        localAreaIdBandwidthPackage:=
model.GetCreateBandwidthPackageLocalAreaIdEnum().CHINESE_MAINLAND
        interflowModeBandwidthPackage:= model.GetCreateBandwidthPackageInterflowModeEnum().AREA
        bandwidthPackagebody := &model.CreateBandwidthPackage{
```

```
BillingMode: model.GetCreateBandwidthPackageBillingModeEnum().E_5,
Bandwidth: int32(5),
ChargeMode: model.GetCreateBandwidthPackageChargeModeEnum().BANDWIDTH,
RemoteArealId: &remoteArealIdBandwidthPackage,
EnterpriseProjectId: &enterpriseProjectIdBandwidthPackage,
ProjectId: "XXX",
Name: "test_bwp_name",
LocalArealId: &localArealIdBandwidthPackage,
InterflowMode: &interflowModeBandwidthPackage,
}
request.Body = &model.CreateBandwidthPackageRequestBody{
    BandwidthPackage: bandwidthPackagebody,
}
response, err := client.CreateBandwidthPackage(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
201	创建带宽包实例成功。

## 错误码

请参见[错误码](#)。

## 4.3.2 查询带宽包列表

### 功能介绍

查询带宽包列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/bandwidth-packages



表 4-82 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-83 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20
enterprise_project_id	否	Array	根据企业项目ID过滤列表。 数组长度：1 - 20
cloud_connection_id	否	Array	根据云连接的ID过滤列表。 数组长度：1 - 20
status	否	Array	根据状态过滤带宽包实例列表。 ACTIVE：表示状态可用。 数组长度：1 - 10
billing_mode	否	Array	根据计费方式过滤带宽包实例列表。 数组长度：1 - 10

参数	是否必选	参数类型	描述
resource_id	否	Array	根据绑定的资源ID过滤带宽包实例列表。 数组长度：1 - 10

## 请求参数

表 4-84 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-85 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
bandwidth_packages	Array of BandwidthPackage objects	带宽包实例列表。 数组长度：0 - 2000

表 4-86 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096

参数	参数类型	描述
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-87 BandwidthPackage

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
project_id	String	实例所属项目ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
resource_id	String	带宽包实例绑定的资源ID。 最小长度：0 最大长度：36
resource_type	String	带宽包实例绑定的资源类型。cloud_connection: 云连接实例。 最小长度：0 最大长度：32
local_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北

参数	参数类型	描述
remote_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
spec_code	String	带宽包实例的规格编码。bandwidth.aftoela：大陆站+国际站南非-拉美东 bandwidth.aftonla：大陆站+国际站南非-拉美北 bandwidth.aftowla：大陆站+国际站南非-拉美西 bandwidth.aptoaf：国际站亚太-南非 bandwidth.aptoap：国际站亚太-拉美东 bandwidth.aptoela：大陆站+国际站亚太-拉美东 bandwidth.aptonla：大陆站+国际站亚太-拉美北 bandwidth.aptowla：大陆站+国际站亚太-拉美西 bandwidth.cmtoaf：国际站中国大陆-南非 bandwidth.cmtoap：国际站中国大陆-亚太 bandwidth.cmtocm：国际站中国大陆-中国大陆 bandwidth.cmtoela：大陆站+国际站中国大陆-拉美东 bandwidth.cmtonla：大陆站+国际站中国大陆-拉美北 bandwidth.cmtowla：大陆站+国际站中国大陆-拉美西 bandwidth.elatoela：大陆站+国际站拉美东-拉美东 bandwidth.elatonla：大陆站+国际站拉美东-拉美北 bandwidth.wlatoela：大陆站+国际站拉美西-拉美东 bandwidth.wlatonla：大陆站+国际站拉美西-拉美北 bandwidth.wlatowla：大陆站+国际站拉美西-拉美西 最小长度：0 最大长度：32
billing_mode	Integer	带宽包实例在大陆站或国际站的计费方式。1：大陆站包周期 2：国际站包周期 3：大陆站按需计费 4：国际站按需计费 5：大陆站按95方式计费 6：国际站按95方式计费
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	带宽包实例的状态。ACTIVE表示状态 最小长度：0 最大长度：36
admin_state_up	Boolean	带宽包实例的管理状态。
order_id	String	带宽包实例的订单ID。 最小长度：0 最大长度：32

参数	参数类型	描述
product_id	String	带宽包实例的产品ID。 最小长度：0 最大长度：36
charge_mode	String	带宽包实例的计费方式。bandwidth是按带宽计费。 最小长度：0 最大长度：64
bandwidth	Integer	带宽包实例中的带宽值。 最小值：2 最大值：999999
interflow_mode	String	互通类型： <ul style="list-style-type: none"><li>Area: 大区互通</li><li>Region: 城域互通</li></ul> 最小长度：0 最大长度：32

表 4-88 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

查询带宽包实例列表

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/bandwidth-packages
```

## 响应示例

状态码：200

查询带宽包实例列表成功。

```
{  
  "request_id": "61126320a1802d5c6444f9d2d76526c2",  
  "bandwidth_packages": [{  
    "bandwidth": 5,  
    "status": "ACTIVE",
```

```
"id": "XXXXX",
"name": "test_bwp_name",
"description": "",
"domain_id": "XXX",
"created_at": "2023-09-20T08:37:15Z",
"updated_at": "2023-09-20T08:37:15Z",
"project_id": "XXX",
"resource_type": "",
"resource_id": "",
"local_area_id": "Chinese-Mainland",
"remote_area_id": "Chinese-Mainland",
"admin_state_up": true,
"charge_mode": "bandwidth",
"billing_mode": "3",
"spec_code": "bandwidth.cmtocm",
"order_id": "",
"product_id": "",
"enterprise_project_id": "0",
"tags": [],
"interflow_mode": "Area"
}],
"page_info": {
  "next_marker": "XXX",
  "previous_marker": "XXX",
  "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListBandwidthPackagesSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBandwidthPackagesRequest request = new ListBandwidthPackagesRequest();
```

```
request.withStatus();
request.withBillingMode();
request.withResourceId();
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withId();
request.withName();
request.withEnterpriseProjectId();
request.withCloudConnectionId();
try {
    ListBandwidthPackagesResponse response = client.listBandwidthPackages(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBandwidthPackagesRequest()
        request.status =
        request.billing_mode =
        request.resource_id =
        request.limit = <limit>
        request.marker = "<marker>"
        request.id =
        request.name =
        request.enterprise_project_id =
        request.cloud_connection_id =
        response = client.list_bandwidth_packages(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询带宽包实例列表成功。

## 错误码

请参见[错误码](#)。



### 4.3.3 查询带宽包实例

#### 功能介绍

查询带宽包实例。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{domain\_id}/ccaas/bandwidth-packages/{id}

表 4-89 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

#### 请求参数

表 4-90 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

#### 响应参数

状态码： 200

表 4-91 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
bandwidth_package	<b>BandwidthPackage</b> object	带宽包实例。

表 4-92 BandwidthPackage

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
project_id	String	实例所属项目ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
resource_id	String	带宽包实例绑定的资源ID。 最小长度：0 最大长度：36
resource_type	String	带宽包实例绑定的资源类型。cloud_connection: 云连接实例。 最小长度：0 最大长度：32
local_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北

参数	参数类型	描述
remote_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
spec_code	String	带宽包实例的规格编码。bandwidth.aftoela：大陆站+国际站南非-拉美东 bandwidth.aftonla：大陆站+国际站南非-拉美北 bandwidth.aftowla：大陆站+国际站南非-拉美西 bandwidth.aptoaf：国际站亚太-南非 bandwidth.aptoap：国际站亚太-亚太 bandwidth.aptoela：大陆站+国际站亚太-拉美东 bandwidth.aptonla：大陆站+国际站亚太-拉美北 bandwidth.aptowla：大陆站+国际站亚太-拉美西 bandwidth.cmtoaf：国际站中国大陆-南非 bandwidth.cmtoap：国际站中国大陆-亚太 bandwidth.cmtocm：国际站中国大陆-中国大陆 bandwidth.cmtoela：大陆站+国际站中国大陆-拉美东 bandwidth.cmtonla：大陆站+国际站中国大陆-拉美北 bandwidth.cmtowla：大陆站+国际站中国大陆-拉美西 bandwidth.elatoela：大陆站+国际站拉美东-拉美东 bandwidth.elatonla：大陆站+国际站拉美东-拉美北 bandwidth.wlatoela：大陆站+国际站拉美西-拉美东 bandwidth.wlatonla：大陆站+国际站拉美西-拉美北 bandwidth.wlatowla：大陆站+国际站拉美西-拉美西 最小长度：0 最大长度：32
billing_mode	Integer	带宽包实例在大陆站或国际站的计费方式。1：大陆站包周期 2：国际站包周期 3：大陆站按需计费 4：国际站按需计费 5：大陆站按95方式计费 6：国际站按95方式计费
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	带宽包实例的状态。ACTIVE表示状态 最小长度：0 最大长度：36
admin_state_up	Boolean	带宽包实例的管理状态。
order_id	String	带宽包实例的订单ID。 最小长度：0 最大长度：32

参数	参数类型	描述
product_id	String	带宽包实例的产品ID。 最小长度：0 最大长度：36
charge_mode	String	带宽包实例的计费方式。bandwidth是按带宽计费。 最小长度：0 最大长度：64
bandwidth	Integer	带宽包实例中的带宽值。 最小值：2 最大值：999999
interflow_mode	String	互通类型： <ul style="list-style-type: none"><li>Area: 大区互通</li><li>Region: 城域互通</li></ul> 最小长度：0 最大长度：32

表 4-93 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

查询带宽包实例详情

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/bandwidth-packages/{id}
```

## 响应示例

状态码：200

查询带宽包实例详情成功。

```
{  
  "bandwidth_package": {  
    "bandwidth": 5,  
    "status": "ACTIVE",  
    "id": "XXXXX",
```

```
"name" : "test_bwp_name",
"description" : "",
"domain_id" : "XXX",
"created_at" : "2023-09-20T08:37:15Z",
"updated_at" : "2023-09-20T08:37:15Z",
"project_id" : "XXX",
"resource_type" : "",
"resource_id" : "",
"local_area_id" : "Chinese-Mainland",
"remote_area_id" : "Chinese-Mainland",
"admin_state_up" : true,
"charge_mode" : "bandwidth",
"billing_mode" : "3",
"spec_code" : "bandwidth.cmtocm",
"order_id" : "",
"product_id" : "",
"enterprise_project_id" : "0",
"tags" : [],
"interflow_mode" : "Area"
},
"request_id" : "39421c95a1d4308d964180f5d51d2f3c"
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ShowBandwidthPackageSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowBandwidthPackageRequest request = new ShowBandwidthPackageRequest();
        try {
            ShowBandwidthPackageResponse response = client.showBandwidthPackage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowBandwidthPackageRequest()
        response = client.show_bandwidth_package(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询带宽包实例详情成功。

## 错误码

请参见[错误码](#)。

## 4.3.4 更新带宽包实例

### 功能介绍

更新带宽包实例。

### 调用方法

请参见[如何调用API](#)。

## URI

PUT /v3/{domain\_id}/ccaas/bandwidth-packages/{id}

表 4-94 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

参数	是否必选	参数类型	描述
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-95 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-96 请求 Body 参数

参数	是否必选	参数类型	描述
bandwidth_package	是	UpdateBandwidthPackage object	更新带宽包实例的请求体

表 4-97 UpdateBandwidthPackage

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
bandwidth	否	Integer	带宽包实例中的带宽值。 最小值：2 最大值：999999
billing_mode	否	Integer	带宽包实例在大陆站或国际站的计费方式： <ul style="list-style-type: none"><li>5：大陆站按95方式计费</li><li>6：国际站按95方式计费</li></ul>



## 响应参数

状态码： 200

表 4-98 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
bandwidth_package	<b>BandwidthPackage</b> object	带宽包实例。

表 4-99 BandwidthPackage

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度： 1 最大长度： 36
project_id	String	实例所属项目ID。
created_at	String	实例创建时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
resource_id	String	带宽包实例绑定的资源ID。 最小长度： 0 最大长度： 36
resource_type	String	带宽包实例绑定的资源类型。 cloud_connection: 云连接实例。 最小长度： 0 最大长度： 32

参数	参数类型	描述
local_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
remote_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
spec_code	String	带宽包实例的规格编码。bandwidth.aftoela：大陆站+国际站南非-拉美东 bandwidth.aftonla：大陆站+国际站南非-拉美北 bandwidth.aftowla：大陆站+国际站南非-拉美西 bandwidth.aptoaf：国际站亚太-南非 bandwidth.aptoap：国际站亚太-亚太 bandwidth.aptoela：大陆站+国际站亚太-拉美东 bandwidth.aptonla：大陆站+国际站亚太-拉美北 bandwidth.aptowla：大陆站+国际站亚太-拉美西 bandwidth.cmtoaf：国际站中国大陆-南非 bandwidth.cmtoap：国际站中国大陆-亚太 bandwidth.cmtocm：国际站中国大陆-中国大陆 bandwidth.cmtoela：大陆站+国际站中国大陆-拉美东 bandwidth.cmtonla：大陆站+国际站中国大陆-拉美北 bandwidth.cmtowla：大陆站+国际站中国大陆-拉美西 bandwidth.elatoela：大陆站+国际站拉美东-拉美东 bandwidth.elatonla：大陆站+国际站拉美东-拉美北 bandwidth.wlatoela：大陆站+国际站拉美西-拉美东 bandwidth.wlatonla：大陆站+国际站拉美西-拉美北 bandwidth.wlatowla：大陆站+国际站拉美西-拉美西 最小长度：0 最大长度：32
billing_mode	Integer	带宽包实例在大陆站或国际站的计费方式。1：大陆站包周期 2：国际站包周期 3：大陆站按需计费 4：国际站按需计费 5：大陆站按95方式计费 6：国际站按95方式计费
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	带宽包实例的状态。ACTIVE表示状态 最小长度：0 最大长度：36
admin_state_up	Boolean	带宽包实例的管理状态。

参数	参数类型	描述
order_id	String	带宽包实例的订单ID。 最小长度：0 最大长度：32
product_id	String	带宽包实例的产品ID。 最小长度：0 最大长度：36
charge_mode	String	带宽包实例的计费方式。bandwidth是按带宽计费。 最小长度：0 最大长度：64
bandwidth	Integer	带宽包实例中的带宽值。 最小值：2 最大值：999999
interflow_mode	String	互通类型： <ul style="list-style-type: none"><li>Area: 大区互通</li><li>Region: 城域互通</li></ul> 最小长度：0 最大长度：32

表 4-100 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

### 更新带宽包实例的带宽值

```
PUT https://{cc_endpoint}/v3/{domain_id}/ccaas/bandwidth-packages/{id}
```

```
{  
  "bandwidth_package": {  
    "bandwidth": 10  
  }  
}
```

## 响应示例

**状态码： 200**

更新带宽包实例成功。

```
{
  "request_id": "d58cc011274d4702642f71318e3e488c",
  "bandwidth_package": {
    "bandwidth": 10,
    "status": "ACTIVE",
    "id": "XXXXX",
    "name": "test_bwp_name",
    "description": "",
    "domain_id": "XXX",
    "created_at": "2023-09-20T08:37:15Z",
    "updated_at": "2023-09-20T09:20:15Z",
    "project_id": "XXX",
    "resource_type": "",
    "resource_id": "",
    "local_area_id": "Chinese-Mainland",
    "remote_area_id": "Chinese-Mainland",
    "admin_state_up": true,
    "charge_mode": "bandwidth",
    "billing_mode": "5",
    "spec_code": "bandwidth.cmtocm",
    "order_id": "",
    "product_id": "",
    "enterprise_project_id": "0",
    "tags": [],
    "interflow_mode": "Area"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新带宽包实例的带宽值

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class UpdateBandwidthPackageSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateBandwidthPackageRequest request = new UpdateBandwidthPackageRequest();
UpdateBandwidthPackageRequestBody body = new UpdateBandwidthPackageRequestBody();
UpdateBandwidthPackage bandwidthPackagebody = new UpdateBandwidthPackage();
bandwidthPackagebody.withBandwidth(10);
body.withBandwidthPackage(bandwidthPackagebody);
request.withBody(body);
try {
    UpdateBandwidthPackageResponse response = client.updateBandwidthPackage(request);
    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

### 更新带宽包实例的带宽值

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateBandwidthPackageRequest()
        bandwidthPackagebody = UpdateBandwidthPackage(
            bandwidth=10
        )
        request.body = UpdateBandwidthPackageRequestBody(
            bandwidth_package=bandwidthPackagebody
        )
        response = client.update_bandwidth_package(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

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

## Go

### 更新带宽包实例的带宽值

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateBandwidthPackageRequest{}
    bandwidthBandwidthPackage := int32(10)
    bandwidthPackagebody := &model.UpdateBandwidthPackage{
        Bandwidth: &bandwidthBandwidthPackage,
    }
    request.Body = &model.UpdateBandwidthPackageRequestBody{
        BandwidthPackage: bandwidthPackagebody,
    }
    response, err := client.UpdateBandwidthPackage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	更新带宽包实例成功。

## 错误码

请参见[错误码](#)。

## 4.3.5 删除带宽包实例

### 功能介绍

删除带宽包实例。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/ccaas/bandwidth-packages/{id}

表 4-101 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 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	用户Token。 最小长度：0 最大长度：2000000

### 响应参数

无

### 请求示例

删除带宽包实例

```
DELETE https://{cc_endpoint}/v3/{domain_id}/ccaas/bandwidth-packages/{id}
```

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteBandwidthPackageSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteBandwidthPackageRequest request = new DeleteBandwidthPackageRequest();
        try {
            DeleteBandwidthPackageResponse response = client.deleteBandwidthPackage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteBandwidthPackageRequest()
        response = client.delete_bandwidth_package(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

```
}  
}
```

## 更多

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

## 状态码

状态码	描述
204	删除带宽包实例成功。

## 错误码

请参见[错误码](#)。

## 4.3.6 创建带宽包标签

### 功能介绍

创建带宽包标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/bandwidth-packages/{id}/tag

表 4-103 路径参数

参数	是否必选	参数类型	描述
id	是	String	资源的Id。 最小长度：32 最大长度：36
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-104 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>2000000</b>

表 4-105 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>Tag</b> objects	包含标签。 数组长度： <b>1 - 20</b>

表 4-106 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 响应参数

状态码： 204

表 4-107 响应 Header 参数

参数	参数类型	描述
x-request-id	-	-

## 请求示例

创建带宽包标签

```
POST /v3/{domain_id}/ccaas/bandwidth-packages/{id}/tag
```

```
{
  "tags" : [ {
    "key" : "key",
    "value" : "value"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

#### 创建带宽包标签

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class TagBandwidthPackageSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        TagBandwidthPackageRequest request = new TagBandwidthPackageRequest();
        TagBandwidthPackageRequestBody body = new TagBandwidthPackageRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key")
                .withValue("value")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            TagBandwidthPackageResponse response = client.tagBandwidthPackage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

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

## Python

### 创建带宽包标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = TagBandwidthPackageRequest()
        listTagsbody = [
            Tag(
                key="key",
                value="value"
            )
        ]
        request.body = TagBandwidthPackageRequestBody(
            tags=listTagsbody
        )
        response = client.tag_bandwidth_package(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 创建带宽包标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
```

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.TagBandwidthPackageRequest{
        valueTags:= "value"
    }
    var listTagsbody = []model.Tag{
        {
            Key: "key",
            Value: &valueTags,
        },
    }
    request.Body = &model.TagBandwidthPackageRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.TagBandwidthPackage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
204	创建带宽包标签成功。

## 错误码

请参见[错误码](#)。

## 4.3.7 删除带宽包标签

### 功能介绍

删除带宽包标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/bandwidth-packages/{id}/untag

表 4-108 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

### 请求参数

表 4-109 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-110 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>Tag</b> objects	包含标签。 数组长度：1 - 20

表 4-111 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 响应参数

状态码： 204

表 4-112 响应 Header 参数

参数	参数类型	描述
x-request-id	-	-

## 请求示例

删除带宽包标签

```
POST /v3/{domain_id}/ccaas/bandwidth-packages/{id}/untag
```

```
{
  "tags": [{
    "key": "key",
    "value": "value"
  }]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

删除带宽包标签

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class UntagBandwidthPackageSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        UntagBandwidthPackageRequest request = new UntagBandwidthPackageRequest();
        UntagBandwidthPackageRequestBody body = new UntagBandwidthPackageRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key")
                .withValue("value")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            UntagBandwidthPackageResponse response = client.untagBandwidthPackage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 删除带宽包标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

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

```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UntagBandwidthPackageRequest()
    listTagsbody = [
        Tag(
            key="key",
            value="value"
        )
    ]
    request.body = UntagBandwidthPackageRequestBody(
        tags=listTagsbody
    )
    response = client.untag_bandwidth_package(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 删除带宽包标签

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UntagBandwidthPackageRequest{
        valueTags:= "value"
        var listTagsbody = []model.Tag{
```

```
{
  Key: "key",
  Value: &valueTags,
},
}
request.Body = &model.UntagBandwidthPackageRequestBody{
  Tags: listTagsbody,
}
response, err := client.UntagBandwidthPackage(request)
if err == nil {
  fmt.Printf("%+v\n", response)
} else {
  fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
204	删除带宽包标签成功。

## 错误码

请参见[错误码](#)。

## 4.3.8 查询带宽包的标签信息

### 功能介绍

查询带宽包的标签信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/bandwidth-packages/tags

表 4-113 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-114 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-115 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
tags	Array of <b>MultivaluedTag</b> objects	带宽包的所有标签。 数组长度：1 - 200

表 4-116 MultivaluedTag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
values	Array of strings	相同键的值列表 数组长度：1 - 20

## 请求示例

查询带宽包标签

GET /v3/{domain\_id}/ccaas/bandwidth-packages/tags

## 响应示例

状态码：200

查询带宽包标签信息成功。

```
{
  "request_id" : "762f22e24aec86c3d61d8db870e87ead",
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1" ]
  }, {
    "key" : "key2",
    "values" : [ "value2" ]
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ListBandwidthPackageTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBandwidthPackageTagsRequest request = new ListBandwidthPackageTagsRequest();
        try {
            ListBandwidthPackageTagsResponse response = client.listBandwidthPackageTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBandwidthPackageTagsRequest()
        response = client.list_bandwidth_package_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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBandwidthPackageTagsRequest{}
    response, err := client.ListBandwidthPackageTags(request)
```

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

## 更多

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

## 状态码

状态码	描述
200	查询带宽包标签信息成功。

## 错误码

请参见[错误码](#)。

## 4.3.9 通过标签过滤带宽包实例

### 功能介绍

通过标签过滤带宽包实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/bandwidth-packages/filter

表 4-117 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-118 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-119 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>MultivaluedTag</b> objects	包含标签。 数组长度：1 - 20

表 4-120 MultivaluedTag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个 unicode字符，格式为大小写字 母，数字，中划线“-”，下划 线“_”，中文。
values	是	Array of strings	相同键的值列表 数组长度：1 - 20

## 响应参数

状态码：200

表 4-121 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	<b>PageInfo</b> object	分页查询页的信息。



参数	参数类型	描述
bandwidth_packages	Array of <a href="#">BandwidthPackage</a> objects	带宽包实例列表。 数组长度：0 - 2000

表 4-122 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-123 BandwidthPackage

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
project_id	String	实例所属项目ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。

参数	参数类型	描述
resource_id	String	带宽包实例绑定的资源ID。 最小长度：0 最大长度：36
resource_type	String	带宽包实例绑定的资源类型。cloud_connection: 云连接实例。 最小长度：0 最大长度：32
local_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
remote_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
spec_code	String	带宽包实例的规格编码。bandwidth.aftoela: 大陆站+国际站南非-拉美东 bandwidth.aftonla: 大陆站+国际站南非-拉美北 bandwidth.aftowla: 大陆站+国际站南非-拉美西 bandwidth.aptoaf: 国际站亚太-南非 bandwidth.aptoap: 国际站亚太-亚太 bandwidth.aptoela: 大陆站+国际站亚太-拉美东 bandwidth.aptonla: 大陆站+国际站亚太-拉美北 bandwidth.aptowla: 大陆站+国际站亚太-拉美西 bandwidth.cmtoaf: 国际站中国大陆-南非 bandwidth.cmtoap: 国际站中国大陆-亚太 bandwidth.cmtoam: 国际站中国大陆-中国大陆 bandwidth.cmtoela: 大陆站+国际站中国大陆-拉美东 bandwidth.cmtonla: 大陆站+国际站中国大陆-拉美北 bandwidth.cmtowla: 大陆站+国际站中国大陆-拉美西 bandwidth.elatoela: 大陆站+国际站拉美东-拉美东 bandwidth.elatonla: 大陆站+国际站拉美东-拉美北 bandwidth.wlatoela: 大陆站+国际站拉美西-拉美东 bandwidth.wlatonla: 大陆站+国际站拉美西-拉美北 bandwidth.wlatowla: 大陆站+国际站拉美西-拉美西 最小长度：0 最大长度：32
billing_mode	Integer	带宽包实例在大陆站或国际站的计费方式。1: 大陆站包周期 2: 国际站包周期 3: 大陆站按需计费 4: 国际站按需计费 5: 大陆站按95方式计费 6: 国际站按95方式计费

参数	参数类型	描述
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	带宽包实例的状态。ACTIVE表示状态 最小长度：0 最大长度：36
admin_state_up	Boolean	带宽包实例的管理状态。
order_id	String	带宽包实例的订单ID。 最小长度：0 最大长度：32
product_id	String	带宽包实例的产品ID。 最小长度：0 最大长度：36
charge_mode	String	带宽包实例的计费方式。bandwidth是按带宽计费。 最小长度：0 最大长度：64
bandwidth	Integer	带宽包实例中的带宽值。 最小值：2 最大值：999999
interflow_mode	String	互通类型： <ul style="list-style-type: none"><li>• Area: 大区互通</li><li>• Region: 城域互通</li></ul> 最小长度：0 最大长度：32

表 4-124 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

### 通过标签过滤带宽包实例

```
POST https://{cc_endpoint}/v3/{domain_id}/ccaas/bandwidth-packages/filter

{
  "tags": [ {
    "key": "key",
    "values": [ "value" ]
  } ]
}
```

## 响应示例

### 状态码： 200

通过标签过滤带宽包实例成功。

```
{
  "request_id": "61126320a1802d5c6444f9d2d76526c2",
  "bandwidth_packages": [ {
    "bandwidth": 5,
    "status": "ACTIVE",
    "id": "XXXXX",
    "name": "test_bwp_name",
    "description": "",
    "domain_id": "XXX",
    "created_at": "2023-09-20T08:37:15Z",
    "updated_at": "2023-09-20T08:37:15Z",
    "project_id": "XXX",
    "resource_type": "",
    "resource_id": "",
    "local_area_id": "Chinese-Mainland",
    "remote_area_id": "Chinese-Mainland",
    "admin_state_up": true,
    "charge_mode": "bandwidth",
    "billing_mode": "3",
    "spec_code": "bandwidth.cmtocm",
    "order_id": "",
    "product_id": "",
    "enterprise_project_id": "0",
    "tags": [ {
      "key": "key1",
      "value": "value1"
    }, {
      "key": "key2",
      "value": "value2"
    } ],
    "interflow_mode": "Area"
  } ],
  "page_info": {
    "next_marker": "XXX",
    "previous_marker": "XXX",
    "current_count": 1
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

通过标签过滤带宽包实例

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListBandwidthPackagesByTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBandwidthPackagesByTagsRequest request = new ListBandwidthPackagesByTagsRequest();
        ListBandwidthPackagesByTagsRequestBody body = new ListBandwidthPackagesByTagsRequestBody();
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("value");
        List<MultivaluedTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new MultivaluedTag()
                .withKey("key")
                .withValues(listTagsValues)
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            ListBandwidthPackagesByTagsResponse response = client.listBandwidthPackagesByTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 通过标签过滤带宽包实例

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
```

```
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBandwidthPackagesByTagsRequest()
        listValuesTags = [
            "value"
        ]
        listTagsbody = [
            MultivaluedTag(
                key="key",
                values=listValuesTags
            )
        ]
        request.body = ListBandwidthPackagesByTagsRequestBody(
            tags=listTagsbody
        )
        response = client.list_bandwidth_packages_by_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

### 通过标签过滤带宽包实例

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListBandwidthPackagesByTagsRequest{}
var listValuesTags = []string{
    "value",
}
var listTagsbody = []model.MultivaluedTag{
    {
        Key: "key",
        Values: listValuesTags,
    },
}
request.Body = &model.ListBandwidthPackagesByTagsRequestBody{
    Tags: listTagsbody,
}
response, err := client.ListBandwidthPackagesByTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
200	通过标签过滤带宽包实例成功。

## 错误码

请参见[错误码](#)。

## 4.3.10 将带宽包实例绑定到云连接实例

### 功能介绍

将带宽包实例绑定到云连接实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/bandwidth-packages/{id}/associate

表 4-125 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-126 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-127 请求 Body 参数

参数	是否必选	参数类型	描述
bandwidth_package	是	<a href="#">AssociateBandwidthPackage</a> object	将带宽包实例关联到资源的详细信息。

表 4-128 AssociateBandwidthPackage

参数	是否必选	参数类型	描述
resource_id	是	String	带宽包实例绑定的资源ID。 最小长度：0 最大长度：36
resource_type	是	String	带宽包实例绑定的资源类型。 cloud_connection: 云连接实例。 最小长度：0 最大长度：32



## 响应参数

状态码： 201

表 4-129 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
bandwidth_package	<b>BandwidthPackage</b> object	带宽包实例。

表 4-130 BandwidthPackage

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度： 1 最大长度： 36
project_id	String	实例所属项目ID。
created_at	String	实例创建时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
resource_id	String	带宽包实例绑定的资源ID。 最小长度： 0 最大长度： 36
resource_type	String	带宽包实例绑定的资源类型。 cloud_connection: 云连接实例。 最小长度： 0 最大长度： 32

参数	参数类型	描述
local_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
remote_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
spec_code	String	带宽包实例的规格编码。bandwidth.aftoela：大陆站+国际站南非-拉美东 bandwidth.aftonla：大陆站+国际站南非-拉美北 bandwidth.aftowla：大陆站+国际站南非-拉美西 bandwidth.aptoaf：国际站亚太-南非 bandwidth.aptoap：国际站亚太-亚太 bandwidth.aptoela：大陆站+国际站亚太-拉美东 bandwidth.aptonla：大陆站+国际站亚太-拉美北 bandwidth.aptowla：大陆站+国际站亚太-拉美西 bandwidth.cmttoaf：国际站中国大陆-南非 bandwidth.cmttoap：国际站中国大陆-亚太 bandwidth.cmtocm：国际站中国大陆-中国大陆 bandwidth.cmttoela：大陆站+国际站中国大陆-拉美东 bandwidth.cmttonla：大陆站+国际站中国大陆-拉美北 bandwidth.cmtowla：大陆站+国际站中国大陆-拉美西 bandwidth.elatoela：大陆站+国际站拉美东-拉美东 bandwidth.elatonla：大陆站+国际站拉美东-拉美北 bandwidth.wlatoela：大陆站+国际站拉美西-拉美东 bandwidth.wlatonla：大陆站+国际站拉美西-拉美北 bandwidth.wlatowla：大陆站+国际站拉美西-拉美西 最小长度：0 最大长度：32
billing_mode	Integer	带宽包实例在大陆站或国际站的计费方式。1：大陆站包周期 2：国际站包周期 3：大陆站按需计费 4：国际站按需计费 5：大陆站按95方式计费 6：国际站按95方式计费
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
status	String	带宽包实例的状态。ACTIVE表示状态 最小长度：0 最大长度：36
admin_state_up	Boolean	带宽包实例的管理状态。

参数	参数类型	描述
order_id	String	带宽包实例的订单ID。 最小长度：0 最大长度：32
product_id	String	带宽包实例的产品ID。 最小长度：0 最大长度：36
charge_mode	String	带宽包实例的计费方式。bandwidth是按带宽计费。 最小长度：0 最大长度：64
bandwidth	Integer	带宽包实例中的带宽值。 最小值：2 最大值：999999
interflow_mode	String	互通类型： <ul style="list-style-type: none"><li>Area: 大区互通</li><li>Region: 城域互通</li></ul> 最小长度：0 最大长度：32

表 4-131 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

将带宽包实例绑定到云连接实例

```
POST https://{cc_endpoint}/v3/{domain_id}/ccaas/bandwidth-packages/{id}/associate
{
  "bandwidth_package": {
    "resource_id": "XXX",
    "resource_type": "cloud_connection"
  }
}
```

## 响应示例

### 状态码： 201

将带宽包实例绑定到云连接实例成功。

```
{
  "bandwidth_package": {
    "bandwidth": 5,
    "status": "ACTIVE",
    "id": "XXXXX",
    "name": "test_bwp_name",
    "description": "",
    "domain_id": "XXX",
    "created_at": "2023-09-20T08:37:15Z",
    "updated_at": "2023-09-20T08:37:15Z",
    "project_id": "XXX",
    "resource_type": "cloud_connection",
    "resource_id": "XXX",
    "local_area_id": "Chinese-Mainland",
    "remote_area_id": "Chinese-Mainland",
    "admin_state_up": true,
    "charge_mode": "bandwidth",
    "billing_mode": "5",
    "spec_code": "bandwidth.cmtocm",
    "order_id": "",
    "product_id": "",
    "enterprise_project_id": "0",
    "tags": [],
    "interflow_mode": "Area"
  },
  "request_id": "39421c95a1d4308d964180f5d51d2f3c"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

将带宽包实例绑定到云连接实例

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class AssociateBandwidthPackageSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        AssociateBandwidthPackageRequest request = new AssociateBandwidthPackageRequest();
        AssociateBandwidthPackageRequestBody body = new AssociateBandwidthPackageRequestBody();
        AssociateBandwidthPackage bandwidthPackagebody = new AssociateBandwidthPackage();

bandwidthPackagebody.withResourceType(AssociateBandwidthPackage.ResourceTypeEnum.fromValue("cloud_connection"))
    .withResourceId("XXX");
body.withBandwidthPackage(bandwidthPackagebody);
request.withBody(body);
try {
    AssociateBandwidthPackageResponse response = client.associateBandwidthPackage(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}
```

## Python

### 将带宽包实例绑定到云连接实例

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = AssociateBandwidthPackageRequest()
        bandwidthPackagebody = AssociateBandwidthPackage(
            resource_type="cloud_connection",
            resource_id="XXX"
        )
        request.body = AssociateBandwidthPackageRequestBody(
            bandwidth_package=bandwidthPackagebody
        )
        response = client.associate_bandwidth_package(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 将带宽包实例绑定到云连接实例

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AssociateBandwidthPackageRequest{}
    bandwidthPackagebody := &model.AssociateBandwidthPackage{
        ResourceType: model.GetAssociateBandwidthPackageResourceTypeEnum().CLOUD_CONNECTION,
        ResourceId: "XXX",
    }
    request.Body = &model.AssociateBandwidthPackageRequestBody{
        BandwidthPackage: bandwidthPackagebody,
    }
    response, err := client.AssociateBandwidthPackage(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
201	将带宽包实例绑定到云连接实例成功。

## 错误码

请参见[错误码](#)。

### 4.3.11 解除带宽包实例与云连接实例的绑定

#### 功能介绍

解除带宽包实例与云连接实例的绑定。

#### 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/{domain\_id}/ccaas/bandwidth-packages/{id}/disassociate

表 4-132 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-133 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-134 请求 Body 参数

参数	是否必选	参数类型	描述
bandwidth_package	是	<b>DisassociateBandwidthPackage</b> object	解关联带宽包实例的详细信息。

表 4-135 DisassociateBandwidthPackage

参数	是否必选	参数类型	描述
resource_id	是	String	带宽包实例绑定的资源ID。 最小长度：0 最大长度：36
resource_type	是	String	带宽包实例绑定的资源类型。 cloud_connection: 云连接实例。 最小长度：0 最大长度：32

## 响应参数

状态码：201

表 4-136 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
bandwidth_package	<b>BandwidthPackage</b> object	带宽包实例。

表 4-137 BandwidthPackage

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。



参数	参数类型	描述
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
project_id	String	实例所属项目ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
resource_id	String	带宽包实例绑定的资源ID。 最小长度：0 最大长度：36
resource_type	String	带宽包实例绑定的资源类型。cloud_connection: 云连接实例。 最小长度：0 最大长度：32
local_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北
remote_area_id	String	本端大区。云连接支持的大区有：  Chinese-Mainland   中国大陆    Asia-Pacific   亚太    Africa   非洲    Western-Latin-America   拉美西    Eastern-Latin-America   拉美东    Northern-Latin-America   拉美北

参数	参数类型	描述
spec_code	String	带宽包实例的规格编码。 bandwidth.aftoela: 大陆站+国际站南非-拉美东 bandwidth.aftonla: 大陆站+国际站南非-拉美北 bandwidth.aftowla: 大陆站+国际站南非-拉美西 bandwidth.aptoaf: 国际站亚太-南非 bandwidth.aptoap: 国际站亚太-亚太 bandwidth.aptoela: 大陆站+国际站亚太-拉美东 bandwidth.aptonla: 大陆站+国际站亚太-拉美北 bandwidth.aptowla: 大陆站+国际站亚太-拉美西 bandwidth.cmtoaf: 国际站中国大陆-南非 bandwidth.cmtoap: 国际站中国大陆-亚太 bandwidth.cmtoam: 国际站中国大陆-中国大陆 bandwidth.cmtoela: 大陆站+国际站中国大陆-拉美东 bandwidth.cmtonla: 大陆站+国际站中国大陆-拉美北 bandwidth.cmtowla: 大陆站+国际站中国大陆-拉美西 bandwidth.elatoela: 大陆站+国际站拉美东-拉美东 bandwidth.elatonla: 大陆站+国际站拉美东-拉美北 bandwidth.wlatoela: 大陆站+国际站拉美西-拉美东 bandwidth.wlatonla: 大陆站+国际站拉美西-拉美北 bandwidth.wlatowla: 大陆站+国际站拉美西-拉美西 最小长度: 0 最大长度: 32
billing_mode	Integer	带宽包实例在大陆站或国际站的计费方式。 1: 大陆站包周期 2: 国际站包周期 3: 大陆站按需计费 4: 国际站按需计费 5: 大陆站按95方式计费 6: 国际站按95方式计费
tags	Array of <b>Tag</b> objects	实例标签。 数组长度: 1 - 20
status	String	带宽包实例的状态。 ACTIVE表示状态 最小长度: 0 最大长度: 36
admin_state_up	Boolean	带宽包实例的管理状态。
order_id	String	带宽包实例的订单ID。 最小长度: 0 最大长度: 32
product_id	String	带宽包实例的产品ID。 最小长度: 0 最大长度: 36

参数	参数类型	描述
charge_mode	String	带宽包实例的计费方式。bandwidth是按带宽计费。 最小长度：0 最大长度：64
bandwidth	Integer	带宽包实例中的带宽值。 最小值：2 最大值：999999
interflow_mode	String	互通类型： <ul style="list-style-type: none"><li>Area: 大区互通</li><li>Region: 城域互通</li></ul> 最小长度：0 最大长度：32

表 4-138 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 请求示例

解除带宽包实例与云连接实例的绑定

```
POST https://{cc_endpoint}/v3/{domain_id}/ccaas/bandwidth-packages/{id}/disassociate
{
  "bandwidth_package": {
    "resource_id": "XXX",
    "resource_type": "cloud_connection"
  }
}
```

## 响应示例

状态码：201

解除带宽包实例与云连接实例的绑定成功。

```
{
  "bandwidth_package": {
    "bandwidth": 5,
    "status": "ACTIVE",
```

```
"id" : "XXXXX",
"name" : "test_bwp_name",
"description" : "",
"domain_id" : "XXX",
"created_at" : "2023-09-20T08:37:15Z",
"updated_at" : "2023-09-20T08:37:15Z",
"project_id" : "XXX",
"resource_type" : "",
"resource_id" : "",
"local_area_id" : "Chinese-Mainland",
"remote_area_id" : "Chinese-Mainland",
"admin_state_up" : true,
"charge_mode" : "bandwidth",
"billing_mode" : "5",
"spec_code" : "bandwidth.cmtocm",
"order_id" : "",
"product_id" : "",
"enterprise_project_id" : "0",
"tags" : [],
"interflow_mode" : "Area"
},
"request_id" : "39421c95a1d4308d964180f5d51d2f3c"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

解除带宽包实例与云连接实例的绑定

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DisassociateBandwidthPackageSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        DisassociateBandwidthPackageRequest request = new DisassociateBandwidthPackageRequest();
        DisassociateBandwidthPackageRequestBody body = new DisassociateBandwidthPackageRequestBody();
        DisassociateBandwidthPackage bandwidthPackagebody = new DisassociateBandwidthPackage();
```

```
bandwidthPackagebody.withResourceType(DisassociateBandwidthPackage.ResourceTypeEnum.fromValue("cloud_connection"))
    .withResourceId("XXX");
body.withBandwidthPackage(bandwidthPackagebody);
request.withBody(body);
try {
    DisassociateBandwidthPackageResponse response = client.disassociateBandwidthPackage(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 解除带宽包实例与云连接实例的绑定

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DisassociateBandwidthPackageRequest()
        bandwidthPackagebody = DisassociateBandwidthPackage(
            resource_type="cloud_connection",
            resource_id="XXX"
        )
        request.body = DisassociateBandwidthPackageRequestBody(
            bandwidth_package=bandwidthPackagebody
        )
        response = client.disassociate_bandwidth_package(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 解除带宽包实例与云连接实例的绑定

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisassociateBandwidthPackageRequest{}
    bandwidthPackagebody := &model.DisassociateBandwidthPackage{
        ResourceType: model.GetDisassociateBandwidthPackageResourceTypeEnum().CLOUD_CONNECTION,
        ResourceId: "XXX",
    }
    request.Body = &model.DisassociateBandwidthPackageRequestBody{
        BandwidthPackage: bandwidthPackagebody,
    }
    response, err := client.DisassociateBandwidthPackage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
201	解除带宽包实例与云连接实例的绑定成功。

## 错误码

请参见[错误码](#)。

## 4.4 域间带宽

### 4.4.1 创建域间带宽实例

#### 功能介绍

创建域间带宽实例。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{domain\_id}/ccaas/inter-region-bandwidths

表 4-139 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

#### 请求参数

表 4-140 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-141 请求 Body 参数

参数	是否必选	参数类型	描述
inter_region_bandwidth	是	CreateInterRegionBandwidth object	创建域间带宽的详情信息。

表 4-142 CreateInterRegionBandwidth

参数	是否必选	参数类型	描述
cloud_connection_id	是	String	资源ID标识符。
bandwidth_package_id	是	String	资源ID标识符。
bandwidth	是	Integer	域间带宽值。 最小值：1 最大值：999999
inter_region_ids	是	Array of strings	域间RegionID。 最小长度：0 最大长度：36 数组长度：2 - 2

## 响应参数

状态码： 201

表 4-143 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
inter_region_bandwidth	<b>InterRegionBandwidth</b> object	域间带宽实例。

表 4-144 InterRegionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。



参数	参数类型	描述
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
cloud_connection_id	String	资源ID标识符。
bandwidth_package_id	String	资源ID标识符。
inter_regions	Array of <b>InterRegion</b> objects	域间实例信息。 数组长度：2 - 2
bandwidth	Integer	域间带宽的值。 最小值：1 最大值：999999

表 4-145 InterRegion

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
local_region_id	String	域间实例本端的RegionID。 最小长度：0 最大长度：36
remote_region_id	String	域间实例对端的RegionID。 最小长度：0 最大长度：36

## 请求示例

创建一个域间带宽实例，并指定本端与对端RegionID

```
POST https://{cc_endpoint}/v3/{domain_id}/ccaas/inter-region-bandwidths
```

```
{
  "inter_region_bandwidth" : {
    "bandwidth" : "5",
    "cloud_connection_id" : "XXX",
    "inter_region_ids" : [ "region_a", "region_b" ],
    "bandwidth_package_id" : "XXX"
  }
}
```

## 响应示例

**状态码： 201**

创建域间带宽实例成功。

```
{
  "inter_region_bandwidth" : {
    "id" : "XXX",
    "name" : "",
    "description" : "",
    "domain_id" : "XXX",
    "created_at" : "2023-09-20T11:41:53Z",
    "updated_at" : "2023-09-20T11:41:53Z",
    "bandwidth_package_id" : "XXX",
    "cloud_connection_id" : "XXX",
    "bandwidth" : 5,
    "inter_regions" : [ {
      "id" : "XXX",
      "project_id" : "",
      "local_region_id" : "region_a",
      "remote_region_id" : "region_b"
    }, {
      "id" : "XXX",
      "project_id" : "",
      "local_region_id" : "region_b",
      "remote_region_id" : "region_a"
    } ]
  },
  "request_id" : "fd403735e22a5213845afbf69425a976"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个域间带宽实例，并指定本端与对端RegionID

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class CreateInterRegionBandwidthSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
CreateInterRegionBandwidthRequest request = new CreateInterRegionBandwidthRequest();
CreateInterRegionBandwidthRequestBody body = new CreateInterRegionBandwidthRequestBody();
List<String> listInterRegionBandwidthInterRegionIds = new ArrayList<>();
listInterRegionBandwidthInterRegionIds.add("region_a");
listInterRegionBandwidthInterRegionIds.add("region_b");
CreateInterRegionBandwidth interRegionBandwidthbody = new CreateInterRegionBandwidth();
interRegionBandwidthbody.withBandwidth(5)
    .withBandwidthPackageId("XXX")
    .withInterRegionIds(listInterRegionBandwidthInterRegionIds)
    .withCloudConnectionId("XXX");
body.withInterRegionBandwidth(interRegionBandwidthbody);
request.withBody(body);
try {
    CreateInterRegionBandwidthResponse response = client.createInterRegionBandwidth(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

创建一个域间带宽实例，并指定本端与对端RegionID

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInterRegionBandwidthRequest()
        listInterRegionIdsInterRegionBandwidth = [
            "region_a",
            "region_b"
        ]
        interRegionBandwidthbody = CreateInterRegionBandwidth(
```

```
        bandwidth=5,
        bandwidth_package_id="XXX",
        inter_region_ids=listInterRegionIdsInterRegionBandwidth,
        cloud_connection_id="XXX"
    )
    request.body = CreateInterRegionBandwidthRequestBody(
        inter_region_bandwidth=interRegionBandwidthbody
    )
    response = client.create_inter_region_bandwidth(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

创建一个域间带宽实例，并指定本端与对端RegionID

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInterRegionBandwidthRequest{
        var listInterRegionIdsInterRegionBandwidth = []string{
            "region_a",
            "region_b",
        }
    }
    interRegionBandwidthbody := &model.CreateInterRegionBandwidth{
        Bandwidth: int32(5),
        BandwidthPackageId: "XXX",
        InterRegionIds: listInterRegionIdsInterRegionBandwidth,
        CloudConnectionId: "XXX",
    }
    request.Body = &model.CreateInterRegionBandwidthRequestBody{
        InterRegionBandwidth: interRegionBandwidthbody,
    }
    response, err := client.CreateInterRegionBandwidth(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

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

## 状态码

状态码	描述
201	创建域间带宽实例成功。

## 错误码

请参见[错误码](#)。

## 4.4.2 查询域间带宽列表

### 功能介绍

查询域间带宽列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/inter-region-bandwidths

表 4-146 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-147 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值： <b>1</b> 最大值： <b>1000</b> 缺省值： <b>1000</b>
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度： <b>1</b> 最大长度： <b>4096</b>
id	否	Array	根据id查询，可查询多个id。 数组长度： <b>1 - 20</b>
enterprise_project_id	否	Array	根据企业项目ID过滤列表。 数组长度： <b>1 - 20</b>
cloud_connection_id	否	Array	根据云连接的ID过滤列表。 数组长度： <b>1 - 20</b>
bandwidth_package_id	否	Array	根据带宽包列表过滤域间带宽实例列表。 数组长度： <b>1 - 10</b>

## 请求参数

表 4-148 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>2000000</b>

## 响应参数

状态码：**200**

表 4-149 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
inter_region_bandwidths	Array of InterRegionBandwidth objects	域间带宽实例列表。 数组长度：0 - 2000

表 4-150 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-151 InterRegionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。

参数	参数类型	描述
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
cloud_connection_id	String	资源ID标识符。
bandwidth_package_id	String	资源ID标识符。
inter_regions	Array of <b>InterRegion</b> objects	域间实例信息。 数组长度：2 - 2
bandwidth	Integer	域间带宽的值。 最小值：1 最大值：999999

表 4-152 InterRegion

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
local_region_id	String	域间实例本端的RegionID。 最小长度：0 最大长度：36
remote_region_id	String	域间实例对端的RegionID。 最小长度：0 最大长度：36

## 请求示例

查询域间带宽实例列表

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/inter-region-bandwidths
```

## 响应示例

状态码：200

查询域间带宽实例列表成功。

```
{
  "request_id": "fd403735e22a5213845afbf69425a976",
  "inter_region_bandwidths": [ {
    "id": "XXX",
    "name": "",
    "description": ""
  }
]
```



```
"domain_id" : "XXX",
"created_at" : "2023-09-20T11:41:53Z",
"updated_at" : "2023-09-20T11:41:53Z",
"bandwidth_package_id" : "XXX",
"cloud_connection_id" : "XXX",
"bandwidth" : 5,
"inter_regions" : [ {
  "id" : "XXX",
  "project_id" : "",
  "local_region_id" : "region_a",
  "remote_region_id" : "region_b"
}, {
  "id" : "XXX",
  "project_id" : "",
  "local_region_id" : "region_b",
  "remote_region_id" : "region_a"
} ]
}],
"page_info" : {
  "previous_marker" : "XXX",
  "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListInterRegionBandwidthsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListInterRegionBandwidthsRequest request = new ListInterRegionBandwidthsRequest();
        request.withBandwidthPackageId();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withId();
    }
}
```

```
request.withEnterpriseProjectId();
request.withCloudConnectionId();
try {
    ListInterRegionBandwidthsResponse response = client.listInterRegionBandwidths(request);
    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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInterRegionBandwidthsRequest()
        request.bandwidth_package_id =
        request.limit = <limit>
        request.marker = "<marker>"
        request.id =
        request.enterprise_project_id =
        request.cloud_connection_id =
        response = client.list_inter_region_bandwidths(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
```

```
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询域间带宽实例列表成功。

## 错误码

请参见[错误码](#)。

### 4.4.3 查询域间带宽实例

#### 功能介绍

查询域间带宽实例。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/ccaas/inter-region-bandwidths/{id}

表 4-153 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-154 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-155 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
inter_region_bandwidth	<b>InterRegionBandwidth</b> object	域间带宽实例。

表 4-156 InterRegionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
cloud_connection_id	String	资源ID标识符。
bandwidth_package_id	String	资源ID标识符。
inter_regions	Array of <a href="#">InterRegion</a> objects	域间实例信息。 数组长度：2 - 2
bandwidth	Integer	域间带宽的值。 最小值：1 最大值：999999

表 4-157 InterRegion

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
local_region_id	String	域间实例本端的RegionID。 最小长度：0 最大长度：36
remote_region_id	String	域间实例对端的RegionID。 最小长度：0 最大长度：36

## 请求示例

查询域间带宽实例详情

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/inter-region-bandwidths/id
```

## 响应示例

状态码： 200

查询域间带宽实例详情成功。

```
{
  "request_id": "fd403735e22a5213845afb69425a976",
  "inter_region_bandwidth": {
    "id": "XXX",
    "name": "",
    "description": "",
    "domain_id": "XXX",
    "created_at": "2023-09-20T11:41:53Z",
    "updated_at": "2023-09-20T11:41:53Z",
    "bandwidth_package_id": "XXX",
    "cloud_connection_id": "XXX",
    "bandwidth": 5,
    "inter_regions": [ {
      "id": "XXX",
      "project_id": "",
      "local_region_id": "region_a",
      "remote_region_id": "region_b"
    }, {
      "id": "XXX",
      "project_id": "",
      "local_region_id": "region_b",
      "remote_region_id": "region_a"
    } ]
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ShowInterRegionBandwidthSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ShowInterRegionBandwidthRequest request = new ShowInterRegionBandwidthRequest();
try {
    ShowInterRegionBandwidthResponse response = client.showInterRegionBandwidth(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInterRegionBandwidthRequest()
        response = client.show_inter_region_bandwidth(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
```

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询域间带宽实例详情成功。

## 错误码

请参见[错误码](#)。

### 4.4.4 更新域间带宽实例

#### 功能介绍

更新域间带宽实例。

#### 调用方法

请参见[如何调用API](#)。



## URI

PUT /v3/{domain\_id}/ccaas/inter-region-bandwidths/{id}

表 4-158 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-159 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-160 请求 Body 参数

参数	是否必选	参数类型	描述
inter_region_bandwidth	是	<a href="#">UpdateInterRegionBandwidth</a> object	更新域间带宽的详情信息。

表 4-161 UpdateInterRegionBandwidth

参数	是否必选	参数类型	描述
bandwidth	是	Integer	域间带宽值。 最小值：1 最大值：999999

## 响应参数

状态码：200

表 4-162 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
inter_region_bandwidth	InterRegionBandwidth object	域间带宽实例。

表 4-163 InterRegionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
cloud_connection_id	String	资源ID标识符。
bandwidth_package_id	String	资源ID标识符。
inter_regions	Array of InterRegion objects	域间实例信息。 数组长度：2 - 2
bandwidth	Integer	域间带宽的值。 最小值：1 最大值：999999

表 4-164 InterRegion

参数	参数类型	描述
id	String	资源ID标识符。

参数	参数类型	描述
project_id	String	实例所属项目ID。
local_region_id	String	域间实例本端的RegionID。 最小长度：0 最大长度：36
remote_region_id	String	域间实例对端的RegionID。 最小长度：0 最大长度：36

## 请求示例

更新域间带宽实例的带宽值

```
PUT https://{cc_endpoint}/v3/{domain_id}/ccaas/inter-region-bandwidths/{id}
{
  "inter_region_bandwidth": {
    "bandwidth": "10"
  }
}
```

## 响应示例

状态码：200

更新域间带宽实例成功。

```
{
  "request_id": "fd403735e22a5213845afbf69425a976",
  "inter_region_bandwidth": {
    "id": "XXX",
    "name": "",
    "description": "",
    "domain_id": "XXX",
    "created_at": "2023-09-20T11:41:53Z",
    "updated_at": "2023-09-20T11:45:53Z",
    "bandwidth_package_id": "XXX",
    "cloud_connection_id": "XXX",
    "bandwidth": 10,
    "inter_regions": [ {
      "id": "XXX",
      "project_id": "",
      "local_region_id": "region_a",
      "remote_region_id": "region_b"
    }, {
      "id": "XXX",
      "project_id": "",
      "local_region_id": "region_b",
      "remote_region_id": "region_a"
    } ]
  }
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

### 更新域间带宽实例的带宽值

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class UpdateInterRegionBandwidthSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInterRegionBandwidthRequest request = new UpdateInterRegionBandwidthRequest();
        UpdateInterRegionBandwidthRequestBody body = new UpdateInterRegionBandwidthRequestBody();
        UpdateInterRegionBandwidth interRegionBandwidthbody = new UpdateInterRegionBandwidth();
        interRegionBandwidthbody.withBandwidth(10);
        body.withInterRegionBandwidth(interRegionBandwidthbody);
        request.withBody(body);
        try {
            UpdateInterRegionBandwidthResponse response = client.updateInterRegionBandwidth(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 更新域间带宽实例的带宽值

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInterRegionBandwidthRequest()
        interRegionBandwidthbody = UpdateInterRegionBandwidth(
            bandwidth=10
        )
        request.body = UpdateInterRegionBandwidthRequestBody(
            inter_region_bandwidth=interRegionBandwidthbody
        )
        response = client.update_inter_region_bandwidth(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 更新域间带宽实例的带宽值

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInterRegionBandwidthRequest{
        interRegionBandwidthbody := &model.UpdateInterRegionBandwidth{
```

```
    Bandwidth: int32(10),
  }
  request.Body = &model.UpdateInterRegionBandwidthRequestBody{
    InterRegionBandwidth: interRegionBandwidthbody,
  }
  response, err := client.UpdateInterRegionBandwidth(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}
```

## 更多

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

## 状态码

状态码	描述
200	更新域间带宽实例成功。

## 错误码

请参见[错误码](#)。

## 4.4.5 删除域间带宽实例

### 功能介绍

删除域间带宽实例。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/ccaas/inter-region-bandwidths/{id}

表 4-165 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

参数	是否必选	参数类型	描述
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-166 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

无

## 请求示例

删除域间带宽实例

```
DELETE https://{cc_endpoint}/v3/{domain_id}/ccaas/inter-region-bandwidths/{id}
```

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteInterRegionBandwidthSolution {

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

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteInterRegionBandwidthRequest request = new DeleteInterRegionBandwidthRequest();
try {
    DeleteInterRegionBandwidthResponse response = client.deleteInterRegionBandwidth(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteInterRegionBandwidthRequest()
        response = client.delete_inter_region_bandwidth(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
204	删除域间带宽实例成功。

## 错误码

请参见[错误码](#)。

## 4.5 云连接路由

## 4.5.1 查询云连接路由条目列表

### 功能介绍

查询云连接路由条目列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/cloud-connection-routes

表 4-167 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-168 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围：1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
cloud_connection_id	否	Array	根据云连接的ID过滤列表。 数组长度：1 - 20

参数	是否必选	参数类型	描述
instance_id	否	Array	根据网络实例ID过滤云连接路由条目列表。 数组长度：1 - 10
region_id	否	String	根据Region ID过滤云连接路由条目列表。 最小长度：0 最大长度：36
id	否	String	根据id查询。 最小长度：0 最大长度：36

## 请求参数

表 4-169 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-170 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	<b>PageInfo</b> object	分页查询页的信息。
cloud_connect ion_routes	Array of <b>CloudConnec tionRoute</b> objects	云连接路由实例列表。 数组长度：0 - 2000

表 4-171 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-172 CloudConnectionRoute

参数	参数类型	描述
id	String	资源ID标识符。
cloud_connection_id	String	资源ID标识符。
domain_id	String	实例所属账号ID。
instance_id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
type	String	路由条目下一跳指向的网络实例的类型。 <ul style="list-style-type: none"><li>• VPC：虚拟私有云。</li><li>• VGW：虚拟网关。</li></ul> 最小长度：2 最大长度：3
destination	String	目的地址。 最小长度：0 最大长度：36

## 请求示例

查询云连接路由列表

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/cloud-connection-routes
```

## 响应示例

**状态码： 200**

查询云连接路由条目列表成功。

```
{
  "request_id": "844d75196c487081d2a32187ea9d3757",
  "cloud_connection_routes": [ {
    "id": "XXX",
    "cloud_connection_id": "XXX",
    "domain_id": "XXX",
    "project_id": "XXX",
    "instance_id": "XXX",
    "type": "vpc",
    "region_id": "cn-north-4",
    "destination": "192.168.1.0/24"
  } ],
  "page_info": {
    "previous_marker": "XXX",
    "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListCloudConnectionRoutesSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCloudConnectionRoutesRequest request = new ListCloudConnectionRoutesRequest();
        request.withInstanceId();
        request.withRegionId("<region_id>");
        request.withId("<id>");
    }
}
```

```
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withCloudConnectionId();
try {
    ListCloudConnectionRoutesResponse response = client.listCloudConnectionRoutes(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCloudConnectionRoutesRequest()
        request.instance_id =
        request.region_id = "<region_id>"
        request.id = "<id>"
        request.limit = <limit>
        request.marker = "<marker>"
        request.cloud_connection_id =
        response = client.list_cloud_connection_routes(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
```

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCloudConnectionRoutesRequest{}
    regionIdRequest := "<region_id>"
    request.RegionId = &regionIdRequest
    idRequest := "<id>"
    request.Id = &idRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListCloudConnectionRoutes(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	查询云连接路由条目列表成功。

## 错误码

请参见[错误码](#)。

## 4.5.2 查询云连接路由条目详情

### 功能介绍

查询云连接路由条目列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/cloud-connection-routes/{id}

表 4-173 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

### 请求参数

表 4-174 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

### 响应参数

状态码：200



表 4-175 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
cloud_connection_route	CloudConnectionRoute object	云连接路由实例。

表 4-176 CloudConnectionRoute

参数	参数类型	描述
id	String	资源ID标识符。
cloud_connection_id	String	资源ID标识符。
domain_id	String	实例所属账号ID。
instance_id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
type	String	路由条目下一跳指向的网络实例的类型。 <ul style="list-style-type: none"><li>• VPC：虚拟私有云。</li><li>• VGW：虚拟网关。</li></ul> 最小长度：2 最大长度：3
destination	String	目的地址。 最小长度：0 最大长度：36

## 请求示例

查询云连接路由详情

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/cloud-connection-routes/{id}
```

## 响应示例

状态码：200

查询云连接路由条目详情成功。

```
{  
  "request_id": "844d75196c487081d2a32187ea9d3757",  
}
```

```
"cloud_connection_route" : {
  "id" : "XXX",
  "cloud_connection_id" : "XXX",
  "domain_id" : "XXX",
  "project_id" : "XXX",
  "instance_id" : "XXX",
  "type" : "vpc",
  "region_id" : "cn-north-4",
  "destination" : "192.168.1.0/24"
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ShowCloudConnectionRoutesSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowCloudConnectionRoutesRequest request = new ShowCloudConnectionRoutesRequest();
        try {
            ShowCloudConnectionRoutesResponse response = client.showCloudConnectionRoutes(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCloudConnectionRoutesRequest()
        response = client.show_cloud_connection_routes(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowCloudConnectionRoutesRequest{}
    response, err := client.ShowCloudConnectionRoutes(request)
```

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

## 更多

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

## 状态码

状态码	描述
200	查询云连接路由条目详情成功。

## 错误码

请参见[错误码](#)。

# 4.6 授权管理

## 4.6.1 创建授权

### 功能介绍

网络实例所属租户授予云连接实例所属租户加载其网络实例的权限。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/authorisations

表 4-177 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-178 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-179 请求 Body 参数

参数	是否必选	参数类型	描述
authorisation	是	CreateAuthorisation object	创建授权的详细信息。

表 4-180 CreateAuthorisation

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
instance_id	是	String	资源ID标识符。
project_id	是	String	实例所属项目ID。
region_id	是	String	RegionID。
cloud_connection_id	是	String	资源ID标识符。
instance_type	是	String	授权网络实例的类型： <ul style="list-style-type: none"><li>vpc：虚拟私有云</li></ul> 最小长度：0 最大长度：255
cloud_connection_domain_id	是	String	被授权云连接实例所属的账户ID。 最小长度：0 最大长度：36

## 响应参数

状态码： 201

表 4-181 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
authorisation	Authorisation object	授权

表 4-182 Authorisation

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
instance_id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
created_at	String	实例创建时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
cloud_connection_id	String	资源ID标识符。
status	String	授权的状态。 最小长度： 0 最大长度： 255
instance_type	String	授权实例的类型。 最小长度： 0 最大长度： 255

参数	参数类型	描述
cloud_connection_domain_id	String	被授权云连接实例所属的账户ID。 最小长度：0 最大长度：36

## 请求示例

创建云连接实例所属租户加载网络实例的授权

```
POST https://{cc_endpoint}/v3/{domain_id}/ccaas/authorisations
```

```
{
  "authorisation": {
    "region_id": "region_id",
    "instance_id": "XXX",
    "instance_type": "vpc",
    "cloud_connection_domain_id": "XXX",
    "cloud_connection_id": "XXX",
    "project_id": "XXX"
  }
}
```

## 响应示例

状态码：201

创建授权成功。

```
{
  "request_id": "6a1865ef10104295db6e3b641284b3ed",
  "authorisation": {
    "id": "XXX",
    "name": "",
    "description": "",
    "domain_id": "XXX",
    "created_at": "2023-09-20T08:03:51Z",
    "updated_at": "2022-09-20T08:03:51Z",
    "project_id": "XXX",
    "region_id": "region_id",
    "instance_type": "vpc",
    "instance_id": "XXX",
    "cloud_connection_domain_id": "XXX",
    "cloud_connection_id": "XXX",
    "status": "authorized"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建云连接实例所属租户加载网络实例的授权

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class CreateAuthorisationSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateAuthorisationRequest request = new CreateAuthorisationRequest();
        CreateAuthorisationRequestBody body = new CreateAuthorisationRequestBody();
        CreateAuthorisation authorisationbody = new CreateAuthorisation();
        authorisationbody.withInstanceId("XXX")
            .withCloudConnectionDomainId("XXX")
            .withProjectId("XXX")
            .withRegionId("region_id")
            .withInstanceType(CreateAuthorisation.InstanceTypeEnum.fromValue("vpc"))
            .withCloudConnectionId("XXX");
        body.withAuthorisation(authorisationbody);
        request.withBody(body);
        try {
            CreateAuthorisationResponse response = client.createAuthorisation(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 创建云连接实例所属租户加载网络实例的授权

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

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



```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateAuthorisationRequest()
    authorisationbody = CreateAuthorisation(
        instance_id="XXX",
        cloud_connection_domain_id="XXX",
        project_id="XXX",
        region_id="region_id",
        instance_type="vpc",
        cloud_connection_id="XXX"
    )
    request.body = CreateAuthorisationRequestBody(
        authorisation=authorisationbody
    )
    response = client.create_authorisation(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 创建云连接实例所属租户加载网络实例的授权

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAuthorisationRequest{}
```

```
authorisationbody := &model.CreateAuthorisation{
    InstanceId: "XXX",
    CloudConnectionDomainId: "XXX",
    ProjectId: "XXX",
    RegionId: "region_id",
    InstanceType: model.GetCreateAuthorisationInstanceTypeEnum().VPC,
    CloudConnectionId: "XXX",
}
request.Body = &model.CreateAuthorisationRequestBody{
    Authorisation: authorisationbody,
}
response, err := client.CreateAuthorisation(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
201	创建授权成功。

## 错误码

请参见[错误码](#)。

## 4.6.2 查询授权列表

### 功能介绍

网络实例所属租户查看其已经授予其他租户的权限。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/ccaas/authorisations

表 4-183 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-184 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20
description	否	Array	根据描述查询，可查询多个描述。 数组长度：1 - 20
cloud_connection_id	否	Array	根据云连接的ID过滤列表。 数组长度：1 - 20
instance_id	否	Array	根据实例ID过滤授权列表。 数组长度：1 - 10

## 请求参数

表 4-185 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-186 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
authorisations	Array of Authorisation objects	授权实例列表。 数组长度：0 - 2000

表 4-187 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-188 Authorisation

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
instance_id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
cloud_connection_id	String	资源ID标识符。
status	String	授权的状态。 最小长度：0 最大长度：255
instance_type	String	授权实例的类型。 最小长度：0 最大长度：255
cloud_connection_domain_id	String	被授权云连接实例所属的账户ID。 最小长度：0 最大长度：36

## 请求示例

查询授权列表

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/authorisations
```

## 响应示例

状态码：200

查询授权列表成功。

```
{  
  "request_id": "bce57c774ed47dd828963e5ec3a0b06d",  
}
```

```
"authorisations" : [ {
  "id" : "XXX",
  "name" : "",
  "description" : "",
  "domain_id" : "XXX",
  "created_at" : "2023-09-20T08:03:51Z",
  "updated_at" : "2023-09-20T08:03:51Z",
  "project_id" : "XXX",
  "region_id" : "region_id",
  "instance_type" : "vpc",
  "instance_id" : "XXX",
  "cloud_connection_domain_id" : "XXX",
  "cloud_connection_id" : "XXX",
  "status" : "authorized"
}],
"page_info" : {
  "previous_marker" : "XXX",
  "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListAuthorisationsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAuthorisationsRequest request = new ListAuthorisationsRequest();
        request.withInstanceld();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withId();
        request.withName();
        request.withDescription();
        request.withCloudConnectionId();
    }
}
```

```
try {
    ListAuthorisationsResponse response = client.listAuthorisations(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAuthorisationsRequest()
        request.instance_id =
        request.limit = <limit>
        request.marker = "<marker>"
        request.id =
        request.name =
        request.description =
        request.cloud_connection_id =
        response = client.list_authorisations(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
```

```
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment 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 := cc.NewCcClient(  
        cc.CcClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListAuthorisationsRequest{}  
    limitRequest := int32(<limit>)  
    request.Limit = &limitRequest  
    markerRequest := "<marker>"  
    request.Marker = &markerRequest  
    response, err := client.ListAuthorisations(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	查询授权列表成功。

## 错误码

请参见[错误码](#)。

### 4.6.3 查询权限列表

#### 功能介绍

云连接实例所属租户查询其可加载其他租户的网络实例权限。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。



## 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/ccaas/permissions

表 4-189 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-190 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围：1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20
description	否	Array	根据描述查询，可查询多个描述。 数组长度：1 - 20

参数	是否必选	参数类型	描述
cloud_connection_id	否	Array	根据云连接的ID过滤列表。 数组长度：1 - 20
instance_id	否	Array	根据实例ID过滤授权列表。 数组长度：1 - 10

## 请求参数

表 4-191 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-192 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
permissions	Array of Permission objects	权限实例列表。 数组长度：0 - 2000

表 4-193 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096

参数	参数类型	描述
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-194 Permission

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
instance_id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
cloud_connection_id	String	资源ID标识符。
status	String	授权的状态。 最小长度：0 最大长度：255
instance_type	String	授权实例的类型。 最小长度：0 最大长度：255
instance_domain_id	String	被授权网络实例所属的账户ID。 最小长度：0 最大长度：36

## 请求示例

### 查询权限列表

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/permissions
```

## 响应示例

### 状态码： 200

查询授权列表成功。

```
{
  "request_id": "bce57c774ed47dd828963e5ec3a0b06d",
  "permissions": [ {
    "id": "XXX",
    "name": "",
    "description": "",
    "domain_id": "XXX",
    "created_at": "2023-09-20T08:03:51Z",
    "project_id": "XXX",
    "region_id": "region_id",
    "instance_type": "vpc",
    "instance_id": "XXX",
    "instance_domain_id": "XXX",
    "cloud_connection_id": "XXX",
    "status": "authorized"
  } ],
  "page_info": {
    "previous_marker": "XXX",
    "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListPermissionsSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListPermissionsRequest request = new ListPermissionsRequest();
request.withInstanceId();
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withId();
request.withName();
request.withDescription();
request.withCloudConnectionId();
try {
    ListPermissionsResponse response = client.listPermissions(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListPermissionsRequest()
        request.instance_id =
        request.limit = <limit>
        request.marker = "<marker>"
        request.id =
        request.name =
        request.description =
        request.cloud_connection_id =
        response = client.list_permissions(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

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

## Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询授权列表成功。

## 错误码

请参见[错误码](#)。

## 4.6.4 更新授权

### 功能介绍

更新授权实例。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{domain\_id}/ccaas/authorisations/{id}

表 4-195 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

### 请求参数

表 4-196 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-197 请求 Body 参数

参数	是否必选	参数类型	描述
authorisation	是	<a href="#">UpdateAuthorisation</a> object	更新授权的详细信息。

表 4-198 UpdateAuthorisation

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255

## 响应参数

状态码：200

表 4-199 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
authorisation	Authorisation object	授权

表 4-200 Authorisation

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
instance_id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。



参数	参数类型	描述
cloud_connection_id	String	资源ID标识符。
status	String	授权的状态。 最小长度：0 最大长度：255
instance_type	String	授权实例的类型。 最小长度：0 最大长度：255
cloud_connection_domain_id	String	被授权云连接实例所属的账户ID。 最小长度：0 最大长度：36

## 请求示例

更新授权实例的描述信息

```
PUT https://{cc_endpoint}/v3/{domain_id}/ccaas/authorisations/{id}
```

```
{  
  "authorisation": {  
    "description": "description"  
  }  
}
```

## 响应示例

状态码：200

更新授权实例成功。

```
{  
  "request_id": "7ef7cd8fc2d5140da1cd59eb2dd25a57",  
  "authorisation": {  
    "id": "XXX",  
    "name": "",  
    "description": "description",  
    "domain_id": "XXX",  
    "created_at": "2023-09-20T08:03:51Z",  
    "updated_at": "2023-09-20T08:13:51Z",  
    "project_id": "XXX",  
    "region_id": "region_id",  
    "instance_type": "vpc",  
    "instance_id": "XXX",  
    "cloud_connection_domain_id": "XXX",  
    "cloud_connection_id": "XXX",  
    "status": "authorized"  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

### 更新授权实例的描述信息

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class UpdateAuthorisationSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAuthorisationRequest request = new UpdateAuthorisationRequest();
        UpdateAuthorisationRequestBody body = new UpdateAuthorisationRequestBody();
        UpdateAuthorisation authorisationbody = new UpdateAuthorisation();
        authorisationbody.withDescription("description");
        body.withAuthorisation(authorisationbody);
        request.withBody(body);
        try {
            UpdateAuthorisationResponse response = client.updateAuthorisation(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 更新授权实例的描述信息

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateAuthorisationRequest()
        authorisationbody = UpdateAuthorisation(
            description="description"
        )
        request.body = UpdateAuthorisationRequestBody(
            authorisation=authorisationbody
        )
        response = client.update_authorisation(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 更新授权实例的描述信息

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAuthorisationRequest{
        descriptionAuthorisation: "description"
    }
```

```
authorisationbody := &model.UpdateAuthorisation{
    Description: &descriptionAuthorisation,
}
request.Body = &model.UpdateAuthorisationRequestBody{
    Authorisation: authorisationbody,
}
response, err := client.UpdateAuthorisation(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	更新授权实例成功。

## 错误码

请参见[错误码](#)。

## 4.6.5 删除授权

### 功能介绍

网络实例所属租户取消授予云连接实例所属租户加载其网络实例的权限。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/ccaas/authorisations/{id}

表 4-201 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

参数	是否必选	参数类型	描述
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-202 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

无

## 请求示例

删除云连接实例所属租户加载网络实例的授权

```
DELETE https://{cc_endpoint}/v3/{domain_id}/ccaas/authorisations/{id}
```

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteAuthorisationSolution {

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

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteAuthorisationRequest request = new DeleteAuthorisationRequest();
try {
    DeleteAuthorisationResponse response = client.deleteAuthorisation(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteAuthorisationRequest()
        response = client.delete_authorisation(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
204	删除授权成功。

## 错误码

请参见[错误码](#)。

## 4.7 中心网络管理

## 4.7.1 创建中心网络

### 功能介绍

创建中心网络。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/gcn/central-networks

表 4-203 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

### 请求参数

表 4-204 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-205 请求 Body 参数

参数	是否必选	参数类型	描述
central_network	是	CreateCentralNetwork object	中心网络。

表 4-206 CreateCentralNetwork

参数	是否必选	参数类型	描述
name	是	String	实例名字。



参数	是否必选	参数类型	描述
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
tags	否	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
enterprise_project_id	否	String	实例所属企业项目ID。 最小长度：1 最大长度：36
policy_document	否	<b>CentralNetworkPolicyDocument</b> object	中心网络策略文档。

表 4-207 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-208 CentralNetworkPolicyDocument

参数	是否必选	参数类型	描述
default_plane	是	String	中心网络默认平面的名字。 缺省值： <b>default-plane</b> 最小长度：1 最大长度：64
planes	是	Array of <b>CentralNetworkPlaneDocument</b> objects	中心网络平面列表。 数组长度：1 - 100

参数	是否必选	参数类型	描述
er_instances	否	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络ER实例列表。 数组长度：1 - 200

表 4-209 CentralNetworkPlaneDocument

参数	是否必选	参数类型	描述
name	是	String	实例名字。
associate_er_tables	否	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	否	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-210 AssociateErTableDocument

参数	是否必选	参数类型	描述
project_id	是	String	实例所属项目ID。
region_id	是	String	RegionID。
enterprise_router_id	是	String	企业路由器的ID。
enterprise_router_table_id	是	String	资源ID标识符。

表 4-211 ExcludeErConnectionDocument

参数	是否必选	参数类型	描述
[数组元素]	否	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-212 AssociateErInstanceDocument

参数	是否必选	参数类型	描述
enterprise_router_id	是	String	企业路由器的ID。
project_id	是	String	实例所属项目ID。
region_id	是	String	RegionID。

## 响应参数

状态码： 202

表 4-213 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
central_network	<b>CentralNetwork</b> object	中心网络。

表 4-214 CentralNetwork

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
created_at	String	实例创建时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。

参数	参数类型	描述
state	String	中心网络状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• UPDATING (处理中)</li><li>• FAILED (失败)</li><li>• CREATING (创建中)</li><li>• DELETING (删除中)</li><li>• DELETED (已删除)</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
planes	Array of <b>CentralNetworkPlane</b> objects	中心网平面列表。 数组长度：0 - 2000
er_instances	Array of <b>CentralNetworkErInstance</b> objects	中心网ER实例列表。 数组长度：1 - 200
connections	Array of <b>CentralNetworkConnectionInfo</b> objects	中心网ER连接列表。 数组长度：0 - 2000
default_plane_id	String	资源ID标识符。

表 4-215 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-216 CentralNetworkPlane

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
associate_er_tables	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-217 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。

表 4-218 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-219 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

表 4-220 CentralNetworkEriInstance

参数	参数类型	描述
id	String	资源ID标识符。
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
asn	Long	网络实例BGP协议的AS号。 最小值：0 最大值：4294967295
site_code	String	站点编码定义

表 4-221 CentralNetworkConnectionInfo

参数	参数类型	描述
id	String	资源ID标识符。
plane_id	String	资源ID标识符。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_size	Integer	带宽值定义，单位Mbps。
connection_type	String	中心网络连接类型定义： <ul style="list-style-type: none"><li>ER-ER (ER-ER Connection)</li><li>ER-GDGW (ER-GDGW Attachment Connection)</li><li>ER-ER_ROUTE_TABLE (ER-ER_ROUTE_TABLE Attachment Connection)</li></ul>
connection_point_pair	Array of <b>ConnectionPoint</b> objects	中心网络连接的两个端点定义，长度固定为2的数组。 数组长度：2 - 2

参数	参数类型	描述
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>

表 4-222 ConnectionPoint

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
site_code	String	站点编码定义
instance_id	String	资源ID标识符。
type	String	中心网络连接点类型定义： <ul style="list-style-type: none"><li>• ER (EnterpriseRouter)</li><li>• GDGW (Global DC Gateway)</li><li>• ER_ROUTE_TABLE (Route Table)</li></ul>

## 请求示例

### 创建中心网络

```
POST /v3/{domain_id}/gcn/central-networks
{
  "central_network": {
    "name": "name"
  }
}
```

## 响应示例

**状态码： 202**

接受创建中心网络。

```
{
  "request_id" : "e9837622a6d4f31e4417b5432f42fafa",
  "central_network" : {
    "id" : "e096c86f-817c-418c-945c-6b1d8860a15d",
    "name" : "name",
    "created_at" : "2023-10-09T06:22:40.856Z",
    "updated_at" : "2023-10-09T06:22:40.856Z",
    "domain_id" : "XXX",
    "state" : "CREATING",
    "enterprise_project_id" : "0"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建中心网络

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class CreateCentralNetworkSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateCentralNetworkRequest request = new CreateCentralNetworkRequest();
        CreateCentralNetworkRequestBody body = new CreateCentralNetworkRequestBody();
        CreateCentralNetwork centralNetworkbody = new CreateCentralNetwork();
        centralNetworkbody.setName("name");
        body.withCentralNetwork(centralNetworkbody);
        request.withBody(body);
        try {
            CreateCentralNetworkResponse response = client.createCentralNetwork(request);
            System.out.println(response.toString());
        }
    }
}
```



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

## Python

### 创建中心网络

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateCentralNetworkRequest()
        centralNetworkbody = CreateCentralNetwork(
            name="name"
        )
        request.body = CreateCentralNetworkRequestBody(
            central_network=centralNetworkbody
        )
        response = client.create_central_network(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 创建中心网络

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateCentralNetworkRequest{}
    centralNetworkbody := &model.CreateCentralNetwork{
        Name: "name",
    }
    request.Body = &model.CreateCentralNetworkRequestBody{
        CentralNetwork: centralNetworkbody,
    }
    response, err := client.CreateCentralNetwork(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
202	接受创建中心网络。

## 错误码

请参见[错误码](#)。

## 4.7.2 查询中心网络列表

### 功能介绍

查询中心网络列表。分页查询使用的参数为marker、limit。limit默认值为0，没有指定marker时返回第一条数据。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/gcn/central-networks

表 4-223 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-224 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
sort_key	否	String	排序字段。 缺省值：updated_at 最小长度：1 最大长度：36
sort_dir	否	Object	指定排序是升序还是降序(asc为升序，desc为降序)。
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20

参数	是否必选	参数类型	描述
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20
state	否	Array	根据状态查询，可查询多个状态。 数组长度：1 - 20
enterprise_project_id	否	Array	根据企业项目ID过滤列表。 数组长度：1 - 20
enterprise_router_id	否	Array	根据ER实例ID过滤列表。 数组长度：1 - 10
attachment_instance_id	否	Array	Attachment实例的ID。 数组长度：1 - 10
global_connection_bandwidth_id	否	Array	根据带宽包ID过滤。 数组长度：1 - 20
connection_id	否	Array	连接的ID。 数组长度：1 - 10

## 请求参数

表 4-225 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-226 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36

参数	参数类型	描述
page_info	<b>PageInfo</b> object	分页查询页的信息。
central_networks	Array of <b>CentralNetwork</b> objects	中心网络列表。 数组长度：0 - 2000

表 4-227 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-228 CentralNetwork

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。

参数	参数类型	描述
state	String	中心网络状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• UPDATING (处理中)</li><li>• FAILED (失败)</li><li>• CREATING (创建中)</li><li>• DELETING (删除中)</li><li>• DELETED (已删除)</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
planes	Array of <b>CentralNetworkPlane</b> objects	中心网平面列表。 数组长度：0 - 2000
er_instances	Array of <b>CentralNetworkErInstance</b> objects	中心网ER实例列表。 数组长度：1 - 200
connections	Array of <b>CentralNetworkConnectionInfo</b> objects	中心网ER连接列表。 数组长度：0 - 2000
default_plane_id	String	资源ID标识符。

表 4-229 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-230 CentralNetworkPlane

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
associate_er_tables	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-231 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。

表 4-232 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-233 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

表 4-234 CentralNetworkEriInstance

参数	参数类型	描述
id	String	资源ID标识符。
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
asn	Long	网络实例BGP协议的AS号。 最小值：0 最大值：4294967295
site_code	String	站点编码定义

表 4-235 CentralNetworkConnectionInfo

参数	参数类型	描述
id	String	资源ID标识符。
plane_id	String	资源ID标识符。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_size	Integer	带宽值定义，单位Mbps。
connection_type	String	中心网络连接类型定义： <ul style="list-style-type: none"><li>ER-ER (ER-ER Connection)</li><li>ER-GDGW (ER-GDGW Attachment Connection)</li><li>ER-ER_ROUTE_TABLE (ER-ER_ROUTE_TABLE Attachment Connection)</li></ul>
connection_point_pair	Array of <b>ConnectionPoint</b> objects	中心网络连接的两个端点定义，长度固定为2的数组。 数组长度：2 - 2



参数	参数类型	描述
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>

表 4-236 ConnectionPoint

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
site_code	String	站点编码定义
instance_id	String	资源ID标识符。
type	String	中心网络连接点类型定义： <ul style="list-style-type: none"><li>• ER (EnterpriseRouter)</li><li>• GDGW (Global DC Gateway)</li><li>• ER_ROUTE_TABLE (Route Table)</li></ul>

## 请求示例

查询中心网络

```
GET /v3/{domain_id}/gcn/central-networks
```

## 响应示例

状态码： 200

查询中心网络列表成功。

```
{
  "request_id" : "0b6ba65808f9d7277f916d44845d271c",
  "page_info" : {
    "current_count" : 1
  },
  "central_networks" : [ {
    "id" : "e096c86f-817c-418c-945c-6b1d8860a15d",
    "name" : "name",
    "created_at" : "2023-10-09T06:22:40.856Z",
    "updated_at" : "2023-10-09T06:22:40.856Z",
    "domain_id" : "XXX",
    "state" : "AVAILABLE",
    "enterprise_project_id" : "0"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListCentralNetworksSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCentralNetworksRequest request = new ListCentralNetworksRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withSortKey("<sort_key>");
        request.withSortDir();
        request.withId();
        request.withName();
        request.withState();
        request.withEnterpriseProjectId();
        request.withEnterpriseRouterId();
        request.withAttachmentInstanceId();
        request.withGlobalConnectionBandwidthId();
        request.withConnectionId();
    }
}
```

```
try {
    ListCentralNetworksResponse response = client.listCentralNetworks(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCentralNetworksRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.sort_key = "<sort_key>"
        request.sort_dir =
        request.id =
        request.name =
        request.state =
        request.enterprise_project_id =
        request.enterprise_router_id =
        request.attachment_instance_id =
        request.global_connection_bandwidth_id =
        request.connection_id =
        response = client.list_central_networks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询中心网络列表成功。

## 错误码

请参见[错误码](#)。

## 4.7.3 查询中心网络详情

### 功能介绍

查询中心网络详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/central-networks/{central\_network\_id}

表 4-237 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

### 请求参数

表 4-238 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

### 响应参数

状态码：200

表 4-239 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network	CentralNetwork object	中心网络。

表 4-240 CentralNetwork

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
state	String	中心网络状态。 <ul style="list-style-type: none"><li>● AVAILABLE (可用)</li><li>● UPDATING (处理中)</li><li>● FAILED (失败)</li><li>● CREATING (创建中)</li><li>● DELETING (删除中)</li><li>● DELETED (已删除)</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
tags	Array of Tag objects	实例标签。 数组长度：1 - 20
planes	Array of CentralNetworkPlane objects	中心网平面列表。 数组长度：0 - 2000

参数	参数类型	描述
er_instances	Array of <b>CentralNetworkErInstance</b> objects	中心网ER实例列表。 数组长度：1 - 200
connections	Array of <b>CentralNetworkConnectionInfo</b> objects	中心网ER连接列表。 数组长度：0 - 2000
default_plane_id	String	资源ID标识符。

表 4-241 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-242 CentralNetworkPlane

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
associate_er_tables	Array of <b>AssociateErTableDocument</b> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	Array of <b>ExcludeErConnectionDocument</b> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-243 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。

表 4-244 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-245 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

表 4-246 CentralNetworkErInstance

参数	参数类型	描述
id	String	资源ID标识符。
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
asn	Long	网络实例BGP协议的AS号。 最小值：0 最大值：4294967295



参数	参数类型	描述
site_code	String	站点编码定义

表 4-247 CentralNetworkConnectionInfo

参数	参数类型	描述
id	String	资源ID标识符。
plane_id	String	资源ID标识符。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_size	Integer	带宽值定义，单位Mbps。
connection_type	String	中心网络连接类型定义： <ul style="list-style-type: none"><li>ER-ER (ER-ER Connection)</li><li>ER-GDGW (ER-GDGW Attachment Connection)</li><li>ER-ER_ROUTE_TABLE (ER-ER_ROUTE_TABLE Attachment Connection)</li></ul>
connection_point_pair	Array of <b>ConnectionPoint</b> objects	中心网络连接的两个端点定义，长度固定为2的数组。 数组长度：2 - 2
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>AVAILABLE (可用)</li><li>CREATING (创建中)</li><li>UPDATING (更新中)</li><li>DELETING (删除中)</li><li>FREEZING (冻结中)</li><li>UNFREEZING (解冻中)</li><li>RECOVERING (恢复中)</li><li>FAILED (失败)</li><li>DELETED (已删除)</li><li>APPROVING (审批中)</li><li>APPROVED (已审批)</li><li>UNAPPROVED (审批未通过)</li></ul>

表 4-248 ConnectionPoint

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
site_code	String	站点编码定义
instance_id	String	资源ID标识符。
type	String	中心网络连接点类型定义： <ul style="list-style-type: none"><li>ER (EnterpriseRouter)</li><li>GDGW (Global DC Gateway)</li><li>ER_ROUTE_TABLE (Route Table)</li></ul>

## 请求示例

查询中心网络详情

```
GET /v3/{domain_id}/gcn/central-networks/{central_network_id}
```

## 响应示例

状态码： 200

查询中心网络详情成功。

```
{
  "request_id" : "ebfebe2205832c9ed7c3d1131041cc0f",
  "central_network" : {
    "id" : "e096c86f-817c-418c-945c-6b1d8860a15d",
    "name" : "name",
    "created_at" : "2023-10-09T06:22:40.856Z",
    "updated_at" : "2023-10-09T06:22:40.856Z",
    "domain_id" : "XXX",
    "state" : "AVAILABLE",
    "enterprise_project_id" : "0"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
```

```
import com.huaweicloud.sdk.cc.v3.model.*;

public class ShowCentralNetworkSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowCentralNetworkRequest request = new ShowCentralNetworkRequest();
        try {
            ShowCentralNetworkResponse response = client.showCentralNetwork(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCentralNetworkRequest()
        response = client.show_central_network(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询中心网络详情成功。

## 错误码

请参见[错误码](#)。

## 4.7.4 更新中心网络详情

### 功能介绍

更新中心网络详情。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{domain\_id}/gcn/central-networks/{central\_network\_id}

表 4-249 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

### 请求参数

表 4-250 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-251 请求 Body 参数

参数	是否必选	参数类型	描述
central_network	是	<b>UpdateCentralNetwork</b> object	更新中心网络的详细信息。

表 4-252 UpdateCentralNetwork

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
tags	否	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20

表 4-253 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个 unicode 字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个 unicode 字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 响应参数

状态码：200

表 4-254 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network	<b>CentralNetwork</b> object	中心网络。

表 4-255 CentralNetwork

参数	参数类型	描述
id	String	资源ID标识符。

参数	参数类型	描述
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
state	String	中心网络状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• UPDATING (处理中)</li><li>• FAILED (失败)</li><li>• CREATING (创建中)</li><li>• DELETING (删除中)</li><li>• DELETED (已删除)</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
planes	Array of <b>CentralNetworkPlane</b> objects	中心网平面列表。 数组长度：0 - 2000
er_instances	Array of <b>CentralNetworkErInstance</b> objects	中心网ER实例列表。 数组长度：1 - 200
connections	Array of <b>CentralNetworkConnectionInfo</b> objects	中心网ER连接列表。 数组长度：0 - 2000
default_plane_id	String	资源ID标识符。

表 4-256 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-257 CentralNetworkPlane

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
associate_er_tables	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-258 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。



表 4-259 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-260 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

表 4-261 CentralNetworkErInstance

参数	参数类型	描述
id	String	资源ID标识符。
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
asn	Long	网络实例BGP协议的AS号。 最小值：0 最大值：4294967295
site_code	String	站点编码定义

表 4-262 CentralNetworkConnectionInfo

参数	参数类型	描述
id	String	资源ID标识符。
plane_id	String	资源ID标识符。
global_connection_bandwidth_id	String	全域互联带宽ID。

参数	参数类型	描述
bandwidth_size	Integer	带宽值定义，单位Mbps。
connection_type	String	中心网络连接类型定义： <ul style="list-style-type: none"> <li>ER-ER (ER-ER Connection)</li> <li>ER-GDGW (ER-GDGW Attachment Connection)</li> <li>ER-ER_ROUTE_TABLE (ER-ER_ROUTE_TABLE Attachment Connection)</li> </ul>
connection_point_pair	Array of <b>ConnectionPoint</b> objects	中心网络连接的两个端点定义，长度固定为2的数组。 数组长度：2 - 2
state	String	中心网络连接状态。 <ul style="list-style-type: none"> <li>AVAILABLE (可用)</li> <li>CREATING (创建中)</li> <li>UPDATING (更新中)</li> <li>DELETING (删除中)</li> <li>FREEZING (冻结中)</li> <li>UNFREEZING (解冻中)</li> <li>RECOVERING (恢复中)</li> <li>FAILED (失败)</li> <li>DELETED (已删除)</li> <li>APPROVING (审批中)</li> <li>APPROVED (已审批)</li> <li>UNAPPROVED (审批未通过)</li> </ul>

表 4-263 ConnectionPoint

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
site_code	String	站点编码定义
instance_id	String	资源ID标识符。

参数	参数类型	描述
type	String	中心网络连接点类型定义： <ul style="list-style-type: none"><li>ER (EnterpriseRouter)</li><li>GDGW (Global DC Gateway)</li><li>ER_ROUTE_TABLE (Route Table)</li></ul>

## 请求示例

更新中心网络详情

```
PUT /v3/{domain_id}/gcn/central-networks/{central_network_id}
```

```
{
  "central_network" : {
    "description" : "new description"
  }
}
```

## 响应示例

状态码： 200

更新中心网络详情成功。

```
{
  "request_id" : "b0f83900a5850b757cc59bf919d90947",
  "central_network" : {
    "id" : "e096c86f-817c-418c-945c-6b1d8860a15d",
    "name" : "name",
    "description" : "new description",
    "created_at" : "2023-10-09T06:22:40.856Z",
    "updated_at" : "2023-10-09T06:27:55.75Z",
    "domain_id" : "XXX",
    "state" : "AVAILABLE",
    "enterprise_project_id" : "0"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新中心网络详情

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;
```

```
public class UpdateCentralNetworkSolution {

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

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

        UpdateCentralNetworkRequest request = new UpdateCentralNetworkRequest();
        UpdateCentralNetworkRequestBody body = new UpdateCentralNetworkRequestBody();
        UpdateCentralNetwork centralNetworkbody = new UpdateCentralNetwork();
        centralNetworkbody.withDescription("new description");
        body.withCentralNetwork(centralNetworkbody);
        request.withBody(body);
        try {
            UpdateCentralNetworkResponse response = client.updateCentralNetwork(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 更新中心网络详情

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = UpdateCentralNetworkRequest()
    centralNetworkbody = UpdateCentralNetwork(
        description="new description"
    )
    request.body = UpdateCentralNetworkRequestBody(
        central_network=centralNetworkbody
    )
    response = client.update_central_network(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 更新中心网络详情

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateCentralNetworkRequest{
        descriptionCentralNetwork: "new description"
    }
    centralNetworkbody := &model.UpdateCentralNetwork{
        Description: &descriptionCentralNetwork,
    }
    request.Body = &model.UpdateCentralNetworkRequestBody{
        CentralNetwork: centralNetworkbody,
    }
    response, err := client.UpdateCentralNetwork(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	更新中心网络详情成功。

## 错误码

请参见[错误码](#)。

## 4.7.5 删除中心网络

### 功能介绍

删除中心网络，请先清除附件后再删除中心网络。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/gcn/central-networks/{central\_network\_id}

表 4-264 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

## 请求参数

表 4-265 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：202

表 4-266 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network	CentralNetwork object	中心网络。

表 4-267 CentralNetwork

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。

参数	参数类型	描述
state	String	中心网络状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• UPDATING (处理中)</li><li>• FAILED (失败)</li><li>• CREATING (创建中)</li><li>• DELETING (删除中)</li><li>• DELETED (已删除)</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
planes	Array of <b>CentralNetworkPlane</b> objects	中心网平面列表。 数组长度：0 - 2000
er_instances	Array of <b>CentralNetworkErInstance</b> objects	中心网ER实例列表。 数组长度：1 - 200
connections	Array of <b>CentralNetworkConnectionInfo</b> objects	中心网ER连接列表。 数组长度：0 - 2000
default_plane_id	String	资源ID标识符。

表 4-268 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。



表 4-269 CentralNetworkPlane

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
associate_er_tables	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-270 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。

表 4-271 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-272 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

表 4-273 CentralNetworkEriInstance

参数	参数类型	描述
id	String	资源ID标识符。
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
asn	Long	网络实例BGP协议的AS号。 最小值：0 最大值：4294967295
site_code	String	站点编码定义

表 4-274 CentralNetworkConnectionInfo

参数	参数类型	描述
id	String	资源ID标识符。
plane_id	String	资源ID标识符。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_size	Integer	带宽值定义，单位Mbps。
connection_type	String	中心网络连接类型定义： <ul style="list-style-type: none"><li>ER-ER (ER-ER Connection)</li><li>ER-GDGW (ER-GDGW Attachment Connection)</li><li>ER-ER_ROUTE_TABLE (ER-ER_ROUTE_TABLE Attachment Connection)</li></ul>
connection_point_pair	Array of <b>ConnectionPoint</b> objects	中心网络连接的两个端点定义，长度固定为2的数组。 数组长度：2 - 2

参数	参数类型	描述
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>

表 4-275 ConnectionPoint

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
site_code	String	站点编码定义
instance_id	String	资源ID标识符。
type	String	中心网络连接点类型定义： <ul style="list-style-type: none"><li>• ER (EnterpriseRouter)</li><li>• GDGW (Global DC Gateway)</li><li>• ER_ROUTE_TABLE (Route Table)</li></ul>

## 请求示例

删除中心网络

```
DELETE /v3/{domain_id}/gcn/central-networks/{central_network_id}
```

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteCentralNetworkSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteCentralNetworkRequest request = new DeleteCentralNetworkRequest();
        try {
            DeleteCentralNetworkResponse response = client.deleteCentralNetwork(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

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

```
# In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteCentralNetworkRequest()
    response = client.delete_central_network(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
202	接受删除中心网络。

## 错误码

请参见[错误码](#)。

## 4.7.6 创建中心网络标签

### 功能介绍

创建中心网络标签。

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/{domain\_id}/gcn/central-networks/{central\_network\_id}/tag

表 4-276 路径参数

参数	是否必选	参数类型	描述
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-277 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-278 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>Tag</b> objects	包含标签。 数组长度：1 - 20

表 4-279 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个 unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个 unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 响应参数

状态码： 204

表 4-280 响应 Header 参数

参数	参数类型	描述
x-request-id	-	-

## 请求示例

创建中心网络标签

```
POST /v3/{domain_id}/gcn/central-networks/{central_network_id}/tag
```

```
{
  "tags": [{
    "key": "key",
    "value": "value"
  }]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

### 创建中心网络标签

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class TagCentralNetworkSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        TagCentralNetworkRequest request = new TagCentralNetworkRequest();
        TagCentralNetworkRequestBody body = new TagCentralNetworkRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key")
                .withValue("value")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            TagCentralNetworkResponse response = client.tagCentralNetwork(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 创建中心网络标签



```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = TagCentralNetworkRequest()
        listTagsbody = [
            Tag(
                key="key",
                value="value"
            )
        ]
        request.body = TagCentralNetworkRequestBody(
            tags=listTagsbody
        )
        response = client.tag_central_network(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 创建中心网络标签

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.TagCentralNetworkRequest{
    valueTags:= "value"
    var listTagsbody = []model.Tag{
        {
            Key: "key",
            Value: &valueTags,
        },
    }
    request.Body = &model.TagCentralNetworkRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.TagCentralNetwork(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
204	创建中心网络标签成功

## 错误码

请参见[错误码](#)。

## 4.7.7 删除中心网络标签

### 功能介绍

删除中心网络标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/gcn/central-networks/{central\_network\_id}/untag

表 4-281 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

## 请求参数

表 4-282 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-283 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>Tag</b> objects	包含标签。 数组长度：1 - 20

表 4-284 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 响应参数

状态码： 204

表 4-285 响应 Header 参数

参数	参数类型	描述
x-request-id	-	-

## 请求示例

删除中心网络标签

```
POST /v3/{domain_id}/gcn/central-networks/{central_network_id}/untag
{
  "tags": [ {
    "key": "key",
    "value": "value"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

删除中心网络标签

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class UntagCentralNetworkSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
UntagCentralNetworkRequest request = new UntagCentralNetworkRequest();
UntagCentralNetworkRequestBody body = new UntagCentralNetworkRequestBody();
List<Tag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new Tag()
        .withKey("key")
        .withValue("value")
);
body.withTags(listbodyTags);
request.withBody(body);
try {
    UntagCentralNetworkResponse response = client.untagCentralNetwork(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 删除中心网络标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UntagCentralNetworkRequest()
        listTagsbody = [
            Tag(
                key="key",
                value="value"
            )
        ]
```

```
]
request.body = UntagCentralNetworkRequestBody(
    tags=listTagsbody
)
response = client.untag_central_network(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 删除中心网络标签

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UntagCentralNetworkRequest{
        valueTags:= "value"
    }
    var listTagsbody = []model.Tag{
        {
            Key: "key",
            Value: &valueTags,
        },
    }
    request.Body = &model.UntagCentralNetworkRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.UntagCentralNetwork(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
204	删除中心网络标签成功

## 错误码

请参见[错误码](#)。

## 4.7.8 查询中心网络的标签信息

### 功能介绍

查询中心网络的标签信息。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/gcn/central-networks/tags

表 4-286 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

## 请求参数

表 4-287 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码： 200

表 4-288 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
tags	Array of <b>MultivaluedTag</b> objects	所有标签。 数组长度： 1 - 200

表 4-289 MultivaluedTag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
values	Array of strings	相同键的值列表 数组长度： 1 - 20

## 请求示例

查询中心网络标签

```
GET /v3/{domain_id}/gcn/central-networks/tags
```

## 响应示例

状态码： 200

查询项目标签信息成功

```
{
  "request_id": "2f68486e542a0bb23ab86a826f909ecf",
  "tags": [ {
    "key": "key1",
    "values": [ "value1" ]
  }, {
    "key": "key2",
    "values": [ "value2" ]
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。



## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ListCentralNetworkTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCentralNetworkTagsRequest request = new ListCentralNetworkTagsRequest();
        try {
            ListCentralNetworkTagsResponse response = client.listCentralNetworkTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListCentralNetworkTagsRequest()
    response = client.list_central_network_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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询项目标签信息成功

## 错误码

请参见[错误码](#)。

## 4.7.9 通过标签过滤中心网络实例

### 功能介绍

通过标签过滤中心网络实例。

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/{domain\_id}/gcn/central-networks/filter

表 4-290 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-291 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000

参数	是否必选	参数类型	描述
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096

## 请求参数

表 4-292 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-293 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>MultivaluedTag</b> objects	包含标签。 数组长度：1 - 20

表 4-294 MultivaluedTag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
values	是	Array of strings	相同键的值列表 数组长度：1 - 20

## 响应参数

状态码： 200

表 4-295 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
page_info	PageInfo object	分页查询页的信息。
central_networks	Array of CentralNetwork objects	中心网络列表。 数组长度： 0 - 2000

表 4-296 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度： 1 最大长度： 4096
previous_marker	String	向前分页标识符。 最小长度： 1 最大长度： 4096
current_count	Integer	当前列表中资源数量。 最小值： 0 最大值： 2000

表 4-297 CentralNetwork

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
created_at	String	实例创建时间。UTC时间格式， yyyy-MM-ddTHH:mm:ss。

参数	参数类型	描述
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
state	String	中心网络状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• UPDATING (处理中)</li><li>• FAILED (失败)</li><li>• CREATING (创建中)</li><li>• DELETING (删除中)</li><li>• DELETED (已删除)</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
tags	Array of <a href="#">Tag</a> objects	实例标签。 数组长度：1 - 20
planes	Array of <a href="#">CentralNetworkPlane</a> objects	中心网平面列表。 数组长度：0 - 2000
er_instances	Array of <a href="#">CentralNetworkErInstance</a> objects	中心网ER实例列表。 数组长度：1 - 200
connections	Array of <a href="#">CentralNetworkConnectionInfo</a> objects	中心网ER连接列表。 数组长度：0 - 2000
default_plane_id	String	资源ID标识符。

表 4-298 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。

参数	参数类型	描述
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-299 CentralNetworkPlane

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
associate_er_tables	Array of <b>AssociateErTableDocument</b> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	Array of <b>ExcludeErConnectionDocument</b> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-300 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。

表 4-301 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <b>AssociateErInstanceDocument</b> objects	中心网络平面管理的ER连接

表 4-302 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

表 4-303 CentralNetworkErInstance

参数	参数类型	描述
id	String	资源ID标识符。
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
asn	Long	网络实例BGP协议的AS号。 最小值： <b>0</b> 最大值： <b>4294967295</b>
site_code	String	站点编码定义

表 4-304 CentralNetworkConnectionInfo

参数	参数类型	描述
id	String	资源ID标识符。
plane_id	String	资源ID标识符。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_size	Integer	带宽值定义，单位Mbps。
connection_type	String	中心网络连接类型定义： <ul style="list-style-type: none"><li>ER-ER (ER-ER Connection)</li><li>ER-GDGW (ER-GDGW Attachment Connection)</li><li>ER-ER_ROUTE_TABLE (ER-ER_ROUTE_TABLE Attachment Connection)</li></ul>



参数	参数类型	描述
connection_point_pair	Array of <b>ConnectionPoint</b> objects	中心网络连接的两个端点定义，长度固定为2的数组。 数组长度：2 - 2
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>

表 4-305 ConnectionPoint

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
site_code	String	站点编码定义
instance_id	String	资源ID标识符。
type	String	中心网络连接点类型定义： <ul style="list-style-type: none"><li>• ER (EnterpriseRouter)</li><li>• GDGW (Global DC Gateway)</li><li>• ER_ROUTE_TABLE (Route Table)</li></ul>

## 请求示例

通过标签过滤中心网络实例

```
POST https://{cc_endpoint}/v3/{domain_id}/gcn/central-networks/filter
```

```
{  
  "tags": [ {
```

```
"key": "key",  
"values": [ "value" ]  
}]  
}
```

## 响应示例

**状态码： 200**

通过标签过滤中心网络实例成功。

```
{  
  "request_id": "0b6ba65808f9d7277f916d44845d271c",  
  "page_info": {  
    "current_count": 1  
  },  
  "central_networks": [ {  
    "id": "e096c86f-817c-418c-945c-6b1d8860a15d",  
    "name": "name",  
    "created_at": "2023-10-09T06:22:40.856Z",  
    "updated_at": "2023-10-09T06:22:40.856Z",  
    "domain_id": "XXX",  
    "state": "AVAILABLE",  
    "enterprise_project_id": "0"  
  } ]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

通过标签过滤中心网络实例

```
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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ListCentralNetworksByTagsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CcClient client = CcClient.newBuilder()  
            .withCredential(auth)
```

```
        .withRegion(CcRegion.valueOf("<YOUR REGION>"))
        .build();
ListCentralNetworksByTagsRequest request = new ListCentralNetworksByTagsRequest();
request.withLimit(<limit>);
request.withMarker("<marker>");
ListCentralNetworksByTagsRequestBody body = new ListCentralNetworksByTagsRequestBody();
List<String> listTagsValues = new ArrayList<>();
listTagsValues.add("value");
List<MultivaluedTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new MultivaluedTag()
        .withKey("key")
        .withValues(listTagsValues)
);
body.withTags(listbodyTags);
request.withBody(body);
try {
    ListCentralNetworksByTagsResponse response = client.listCentralNetworksByTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 通过标签过滤中心网络实例

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCentralNetworksByTagsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        listValuesTags = [
            "value"
        ]
        listTagsbody = [
            MultivaluedTag(
```

```
        key="key",
        values=listValuesTags
    )
]
request.body = ListCentralNetworksByTagsRequestBody(
    tags=listTagsbody
)
response = client.list_central_networks_by_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

### 通过标签过滤中心网络实例

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCentralNetworksByTagsRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    var listValuesTags = []string{
        "value",
    }
    var listTagsbody = []model.MultivaluedTag{
        {
            Key: "key",
            Values: listValuesTags,
        },
    }
    request.Body = &model.ListCentralNetworksByTagsRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.ListCentralNetworksByTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```

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

## 更多

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

## 状态码

状态码	描述
200	通过标签过滤中心网络实例成功。

## 错误码

请参见[错误码](#)。

## 4.7.10 创建一个新版本的中心网络策略

### 功能介绍

创建一份只读的中心网络的策略。如果您有策略文档内容改动，需要基于此版本重新创建一个新版本的策略。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/policies

表 4-306 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

## 请求参数

表 4-307 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-308 请求 Body 参数

参数	是否必选	参数类型	描述
central_network_policy_document	是	<b>CentralNetworkPolicyDocument</b> object	中心网络策略文档。

表 4-309 CentralNetworkPolicyDocument

参数	是否必选	参数类型	描述
default_plane	是	String	中心网络默认平面的名字。 缺省值： <b>default-plane</b> 最小长度：1 最大长度：64
planes	是	Array of <b>CentralNetworkPlaneDocument</b> objects	中心网络平面列表。 数组长度：1 - 100
er_instances	否	Array of <b>AssociateErInstanceDocument</b> objects	中心网络ER实例列表。 数组长度：1 - 200

表 4-310 CentralNetworkPlaneDocument

参数	是否必选	参数类型	描述
name	是	String	实例名字。

参数	是否必选	参数类型	描述
associate_er_tables	否	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	否	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-311 AssociateErTableDocument

参数	是否必选	参数类型	描述
project_id	是	String	实例所属项目ID。
region_id	是	String	RegionID。
enterprise_router_id	是	String	企业路由器的ID。
enterprise_router_table_id	是	String	资源ID标识符。

表 4-312 ExcludeErConnectionDocument

参数	是否必选	参数类型	描述
[数组元素]	否	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-313 AssociateErInstanceDocument

参数	是否必选	参数类型	描述
enterprise_router_id	是	String	企业路由器的ID。
project_id	是	String	实例所属项目ID。
region_id	是	String	RegionID。

## 响应参数

状态码： 201

表 4-314 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network_policy	<b>CentralNetworkPolicy</b> object	中心网络策略详情。

表 4-315 CentralNetworkPolicy

参数	参数类型	描述
id	String	资源ID标识符。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
state	String	中心网络策略状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CANCELING (取消中)</li><li>• APPLYING (应用中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li></ul>
central_network_id	String	中心网络ID。
document_template_version	String	文档模板版本。 <ul style="list-style-type: none"><li>• 2022.08.30 (2022.08.30)</li></ul>
is_applied	Boolean	是否被应用。
version	Integer	中心网络策略的版本。 最小值：1 最大值：4096
document	<b>CentralNetworkPolicyDocument</b> object	中心网络策略文档。



表 4-316 CentralNetworkPolicyDocument

参数	参数类型	描述
default_plane	String	中心网络默认平面的名字。 缺省值： <b>default-plane</b> 最小长度： <b>1</b> 最大长度： <b>64</b>
planes	Array of <a href="#">CentralNetworkPlaneDocument</a> objects	中心网络平面列表。 数组长度： <b>1 - 100</b>
er_instances	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络ER实例列表。 数组长度： <b>1 - 200</b>

表 4-317 CentralNetworkPlaneDocument

参数	参数类型	描述
name	String	实例名字。
associate_er_tables	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度： <b>0 - 100</b>
exclude_er_connections	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度： <b>0 - 100</b>

表 4-318 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。

表 4-319 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-320 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

## 请求示例

### 创建中心网络策略

```
POST /v3/{domain_id}/gcn/central-network/{central_network_id}/policies
```

```
{
  "central_network_policy_document" : {
    "default_plane" : "default-plane",
    "planes" : [ {
      "name" : "default-plane"
    } ],
    "er_instances" : [ {
      "enterprise_router_id" : "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
      "project_id" : "XXX",
      "region_id" : "cn-north-4"
    } ]
  }
}
```

## 响应示例

### 状态码： 201

创建中心网络策略成功。

```
{
  "request_id" : "76bfd2b0d492d0e479fbbd3981869c12",
  "central_network_policy" : {
    "id" : "f03478aa-3975-4ca7-9fb2-b49428a01636",
    "created_at" : "2023-10-09T06:22:40.857Z",
    "domain_id" : "XXX",
    "state" : "AVAILABLE",
    "central_network_id" : "e096c86f-817c-418c-945c-6b1d8860a15d",
    "document_template_version" : "2022.08.30",
    "is_applied" : true,
    "version" : 1,
    "document" : {
      "default_plane" : "default-plane",

```

```
"planes": [{
  "name": "default-plane",
  "associate_er_tables": [{
    "project_id": "XXX",
    "region_id": "cn-north-4",
    "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
    "enterprise_router_table_id": "cc542128-5c2d-402a-8960-53bb2ed9484e"
  }]
}],
"er_instances": [{
  "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
  "project_id": "XXX",
  "region_id": "cn-north-4"
}]
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

#### 创建中心网络策略

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class CreateCentralNetworkPolicySolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateCentralNetworkPolicyRequest request = new CreateCentralNetworkPolicyRequest();
        CreateCentralNetworkPolicyRequestBody body = new CreateCentralNetworkPolicyRequestBody();
        List<AssociateErInstanceDocument> listCentralNetworkPolicyDocumentErInstances = new
        ArrayList<>();
        listCentralNetworkPolicyDocumentErInstances.add(
            new AssociateErInstanceDocument()
                .withEnterpriseRouterId("395b0884-aab4-4bf0-8cb8-7f2da26708dd")
                .withRegionId("cn-north-4")
        );
    }
}
```

```
        .withProjectId("XXX")
    );
    List<CentralNetworkPlaneDocument> listCentralNetworkPolicyDocumentPlanes = new ArrayList<>();
    listCentralNetworkPolicyDocumentPlanes.add(
        new CentralNetworkPlaneDocument()
            .withName("default-plane")
    );
    CentralNetworkPolicyDocument centralNetworkPolicyDocumentbody = new
CentralNetworkPolicyDocument();
    centralNetworkPolicyDocumentbody.withDefaultPlane("default-plane")
        .withPlanes(listCentralNetworkPolicyDocumentPlanes)
        .withErInstances(listCentralNetworkPolicyDocumentErInstances);
    body.withCentralNetworkPolicyDocument(centralNetworkPolicyDocumentbody);
    request.withBody(body);
    try {
        CreateCentralNetworkPolicyResponse response = client.createCentralNetworkPolicy(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

### 创建中心网络策略

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateCentralNetworkPolicyRequest()
        listErInstancesCentralNetworkPolicyDocument = [
            AssociateErInstanceDocument(
                enterprise_router_id="395b0884-aab4-4bf0-8cb8-7f2da26708dd",
                region_id="cn-north-4",
                project_id="XXX"
            )
        ]
        listPlanesCentralNetworkPolicyDocument = [
            CentralNetworkPlaneDocument(
```

```
        name="default-plane"
    )
]
centralNetworkPolicyDocumentbody = CentralNetworkPolicyDocument(
    default_plane="default-plane",
    planes=listPlanesCentralNetworkPolicyDocument,
    er_instances=listErInstancesCentralNetworkPolicyDocument
)
request.body = CreateCentralNetworkPolicyRequestBody(
    central_network_policy_document=centralNetworkPolicyDocumentbody
)
response = client.create_central_network_policy(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 创建中心网络策略

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build()
    )

    request := &model.CreateCentralNetworkPolicyRequest{}
    var listErInstancesCentralNetworkPolicyDocument = []model.AssociateErInstanceDocument{
        {
            EnterpriseRouterId: "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
            RegionId: "cn-north-4",
            ProjectId: "XXX",
        },
    }
    var listPlanesCentralNetworkPolicyDocument = []model.CentralNetworkPlaneDocument{
        {
            Name: "default-plane",
        },
    }
    centralNetworkPolicyDocumentbody := &model.CentralNetworkPolicyDocument{
        DefaultPlane: "default-plane",
        Planes: listPlanesCentralNetworkPolicyDocument,
    }
}
```

```
ErInstances: &listErInstancesCentralNetworkPolicyDocument,
}
request.Body = &model.CreateCentralNetworkPolicyRequestBody{
    CentralNetworkPolicyDocument: centralNetworkPolicyDocumentbody,
}
response, err := client.CreateCentralNetworkPolicy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
201	创建中心网络策略成功。

## 错误码

请参见[错误码](#)。

## 4.7.11 查询所有版本的中心网络策略列表

### 功能介绍

查询所有版本的中心网络策略列表。分页查询使用的参数为marker、limit。limit默认值为0，没有指定marker时返回第一条数据。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/policies

表 4-321 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

参数	是否必选	参数类型	描述
central_network_id	是	String	中心网络的ID。 最小长度： <b>36</b> 最大长度： <b>36</b>

表 4-322 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值： <b>1</b> 最大值： <b>1000</b> 缺省值： <b>1000</b>
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度： <b>1</b> 最大长度： <b>4096</b>
sort_key	否	String	排序字段。 缺省值： <b>updated_at</b> 最小长度： <b>1</b> 最大长度： <b>36</b>
sort_dir	否	Object	指定排序是升序还是降序(asc为升序，desc为降序)。
id	否	Array	根据id查询，可查询多个id。 数组长度： <b>1 - 20</b>
state	否	Array	根据状态查询，可查询多个状态。 数组长度： <b>1 - 20</b>
version	否	Array	根据版本查询，可查询多个名字。 数组长度： <b>1 - 20</b>
is_applied	否	Boolean	是否被应用。

## 请求参数

表 4-323 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-324 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
central_network_policies	Array of CentralNetworkPolicy objects	中心网络策略列表。 数组长度：0 - 2000

表 4-325 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000



表 4-326 CentralNetworkPolicy

参数	参数类型	描述
id	String	资源ID标识符。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
state	String	中心网络策略状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CANCELING (取消中)</li><li>• APPLYING (应用中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li></ul>
central_network_id	String	中心网络ID。
document_template_version	String	文档模板版本。 <ul style="list-style-type: none"><li>• 2022.08.30 (2022.08.30)</li></ul>
is_applied	Boolean	是否被应用。
version	Integer	中心网络策略的版本。 最小值：1 最大值：4096
document	<a href="#">CentralNetworkPolicyDocument</a> object	中心网络策略文档。

表 4-327 CentralNetworkPolicyDocument

参数	参数类型	描述
default_plane	String	中心网络默认平面的名字。 缺省值： <b>default-plane</b> 最小长度：1 最大长度：64
planes	Array of <a href="#">CentralNetworkPlaneDocument</a> objects	中心网络平面列表。 数组长度：1 - 100

参数	参数类型	描述
er_instances	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络ER实例列表。 数组长度：1 - 200

表 4-328 CentralNetworkPlaneDocument

参数	参数类型	描述
name	String	实例名字。
associate_er_tables	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-329 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。

表 4-330 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-331 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

## 请求示例

查询中心网络策略列表

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/policies
```

## 响应示例

状态码： 200

查询中心网络策略列表成功。

```
{
  "request_id": "76bfd2b0d492d0e479fbbd3981869c12",
  "page_info": {
    "current_count": 1
  },
  "central_network_policies": [ {
    "id": "f03478aa-3975-4ca7-9fb2-b49428a01636",
    "created_at": "2023-10-09T06:22:40.857Z",
    "domain_id": "XXX",
    "state": "AVAILABLE",
    "central_network_id": "e096c86f-817c-418c-945c-6b1d8860a15d",
    "document_template_version": "2022.08.30",
    "is_applied": true,
    "version": 1,
    "document": {
      "default_plane": "default-plane",
      "planes": [ {
        "name": "default-plane",
        "associate_er_tables": [ {
          "project_id": "XXX",
          "region_id": "cn-north-4",
          "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
          "enterprise_router_table_id": "cc542128-5c2d-402a-8960-53bb2ed9484e"
        }
      ]
    }
  },
  "er_instances": [ {
    "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
    "project_id": "XXX",
    "region_id": "cn-north-4"
  }
]
}
```

## SDK 代码示例

SDK代码示例如下。

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListCentralNetworkPoliciesSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCentralNetworkPoliciesRequest request = new ListCentralNetworkPoliciesRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withSortKey("<sort_key>");
        request.withSortDir();
        request.withId();
        request.withState();
        request.withVersion();
        request.withIsApplied(<is_applied>);
        try {
            ListCentralNetworkPoliciesResponse response = client.listCentralNetworkPolicies(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCentralNetworkPoliciesRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.sort_key = "<sort_key>"
        request.sort_dir =
        request.id =
        request.state =
        request.version =
        request.is_applied = <IsApplied>
        response = client.list_central_network_policies(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCentralNetworkPoliciesRequest{}
    limitRequest:= int32(<limit>)
```

```
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
sortKeyRequest:= "<sort_key>"
request.SortKey = &sortKeyRequest
isAppliedRequest:= <is_applied>
request.IsApplied = &isAppliedRequest
response, err := client.ListCentralNetworkPolicies(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
200	查询中心网络策略列表成功。

## 错误码

请参见[错误码](#)。

## 4.7.12 应用中心网络策略

### 功能介绍

应用中心网络策略。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/policies/{policy\_id}/apply

表 4-332 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

参数	是否必选	参数类型	描述
policy_id	是	String	网络策略ID。 最小长度：36 最大长度：36
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

## 请求参数

表 4-333 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：202

表 4-334 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network_policy	<b>CentralNetworkPolicy</b> object	中心网络策略详情。
central_network_policy_change_set	Array of <b>CentralNetworkElementChange</b> objects	网络策略变化列表。

表 4-335 CentralNetworkPolicy

参数	参数类型	描述
id	String	资源ID标识符。
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
domain_id	String	实例所属账号ID。
state	String	中心网络策略状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CANCELING (取消中)</li><li>• APPLYING (应用中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li></ul>
central_network_id	String	中心网络ID。
document_template_version	String	文档模板版本。 <ul style="list-style-type: none"><li>• 2022.08.30 (2022.08.30)</li></ul>
is_applied	Boolean	是否被应用。
version	Integer	中心网络策略的版本。 最小值：1 最大值：4096
document	<b>CentralNetworkPolicyDocument</b> object	中心网络策略文档。

表 4-336 CentralNetworkPolicyDocument

参数	参数类型	描述
default_plane	String	中心网络默认平面的名字。 缺省值： <b>default-plane</b> 最小长度：1 最大长度：64
planes	Array of <b>CentralNetworkPlaneDocument</b> objects	中心网络平面列表。 数组长度：1 - 100



参数	参数类型	描述
er_instances	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络ER实例列表。 数组长度：1 - 200

表 4-337 CentralNetworkPlaneDocument

参数	参数类型	描述
name	String	实例名字。
associate_er_tables	Array of <a href="#">AssociateErTableDocument</a> objects	关联的中心网络ER实例列表。 数组长度：0 - 100
exclude_er_connections	Array of <a href="#">ExcludeErConnectionDocument</a> objects	当自动连接所有ER实例时，排除中心网络的ER实例的连接。 数组长度：0 - 100

表 4-338 AssociateErTableDocument

参数	参数类型	描述
project_id	String	实例所属项目ID。
region_id	String	RegionID。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_table_id	String	资源ID标识符。

表 4-339 ExcludeErConnectionDocument

参数	参数类型	描述
[数组元素]	Array of <a href="#">AssociateErInstanceDocument</a> objects	中心网络平面管理的ER连接

表 4-340 AssociateErInstanceDocument

参数	参数类型	描述
enterprise_router_id	String	企业路由器的ID。
project_id	String	实例所属项目ID。
region_id	String	RegionID。

表 4-341 CentralNetworkElementChange

参数	参数类型	描述
operation_id	String	实例状态。 <ul style="list-style-type: none"><li>• CreateCentralNetworkPlane: 新增中心网络平面</li><li>• DeleteCentralNetworkPlane: 移除中心网络平面</li><li>• UpdateCentralNetworkPlane: 更新中心网络平面</li><li>• CreateCentralNetworkErInstance: 新增中心网络ER实例</li><li>• DeleteCentralNetworkErInstance: 移除中心网络ER实例</li><li>• CreateCentralNetworkErConnection: 新增中心网络ER连接</li><li>• DeleteCentralNetworkErConnection: 移除中心网络ER连接</li><li>• CreateCentralNetworkErTable: 新增中心网络ER路由表</li><li>• DeleteCentralNetworkErTable: 移除中心网络ER路由表</li></ul>

## 请求示例

应用中心网络策略

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/policies
```

## 响应示例

状态码： 202

接受应用中心网络策略的请求。

```
{  
  "request_id" : "edb137a2c46c5bda0409833359bb649b",  
  "central_network_policy" : {
```

```
"id": "ff51f460-4bbe-4385-b2c4-efbe3318076f",
"created_at": "2023-10-09T07:00:33.663Z",
"domain_id": "XXX",
"state": "APPLYING",
"central_network_id": "e096c86f-817c-418c-945c-6b1d8860a15d",
"document_template_version": "2022.08.30",
"is_applied": false,
"version": 2,
"document": {
  "default_plane": "default-plane",
  "planes": [ {
    "name": "default-plane",
    "associate_er_tables": [ {
      "project_id": "XXX",
      "region_id": "cn-north-4",
      "enterprise_router_id": "c73b26b7-33f0-438d-b440-8e87dfe6fef9",
      "enterprise_router_table_id": "c0d51f20-0313-40f7-a74e-9dccb5da21c0"
    } ]
  } ],
  "er_instances": [ {
    "enterprise_router_id": "c73b26b7-33f0-438d-b440-8e87dfe6fef9",
    "project_id": "XXX",
    "region_id": "cn-north-4"
  } ]
} ],
"central_network_policy_change_set": [ {
  "operation_id": "UpdateCentralNetworkPlane",
  "original_central_network_plane": {
    "name": "default-plane",
    "is_default": true,
    "associate_er_tables": [ {
      "project_id": "XXX",
      "region_id": "cn-north-4",
      "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
      "enterprise_router_table_id": "cc542128-5c2d-402a-8960-53bb2ed9484e"
    } ]
  },
  "newest_central_network_plane": {
    "name": "default-plane",
    "is_default": true,
    "associate_er_tables": [ {
      "project_id": "XXX",
      "region_id": "cn-north-4",
      "enterprise_router_id": "c73b26b7-33f0-438d-b440-8e87dfe6fef9",
      "enterprise_router_table_id": "c0d51f20-0313-40f7-a74e-9dccb5da21c0"
    } ]
  }
} ], {
  "operation_id": "CreateCentralNetworkErInstance",
  "create_central_network_er_instance": {
    "enterprise_router_id": "c73b26b7-33f0-438d-b440-8e87dfe6fef9",
    "project_id": "XXX",
    "region_id": "cn-north-4"
  }
}, {
  "operation_id": "DeleteCentralNetworkErInstance",
  "delete_central_network_er_instance": {
    "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
    "project_id": "XXX",
    "region_id": "cn-north-4"
  }
}, {
  "operation_id": "CreateCentralNetworkErTable",
  "central_network_plane_name": "default-plane",
  "project_id": "XXX",
  "region_id": "cn-north-4",
  "enterprise_router_id": "c73b26b7-33f0-438d-b440-8e87dfe6fef9",
  "enterprise_router_table_id": "c0d51f20-0313-40f7-a74e-9dccb5da21c0"
}
```

```
}, {  
  "operation_id" : "DeleteCentralNetworkErTable",  
  "central_network_plane_name" : "default-plane",  
  "project_id" : "XXX",  
  "region_id" : "cn-north-4",  
  "enterprise_router_id" : "395b0884-aab4-4bf0-8cb8-7f2da26708dd",  
  "enterprise_router_table_id" : "cc542128-5c2d-402a-8960-53bb2ed9484e"  
}]  
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
public class ApplyCentralNetworkPolicySolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CcClient client = CcClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ApplyCentralNetworkPolicyRequest request = new ApplyCentralNetworkPolicyRequest();  
        try {  
            ApplyCentralNetworkPolicyResponse response = client.applyCentralNetworkPolicy(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

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

## Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ApplyCentralNetworkPolicyRequest{}
    response, err := client.ApplyCentralNetworkPolicy(request)
```

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

## 更多

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

## 状态码

状态码	描述
202	接受应用中心网络策略的请求。

## 错误码

请参见[错误码](#)。

## 4.7.13 删除中心网络策略版本

### 功能介绍

删除中心网络策略版本。您无法删除正在被应用的中心策略。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/policies/{policy\_id}

表 4-342 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
policy_id	是	String	网络策略ID。 最小长度：36 最大长度：36

参数	是否必选	参数类型	描述
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

## 请求参数

表 4-343 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：204

表 4-344 响应 Header 参数

参数	参数类型	描述
x-request-id	-	-

## 请求示例

删除中心网络策略

```
DELETE /v3/{domain_id}/gcn/central-network/{central_network_id}/policies/{policy_id}
```

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteCentralNetworkPolicySolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteCentralNetworkPolicyRequest request = new DeleteCentralNetworkPolicyRequest();
        try {
            DeleteCentralNetworkPolicyResponse response = client.deleteCentralNetworkPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```



```
request = DeleteCentralNetworkPolicyRequest()
response = client.delete_central_network_policy(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
204	删除中心网络策略成功

## 错误码

请参见[错误码](#)。

## 4.7.14 查询中心网络策略变化集

### 功能介绍

查询与当前应用中心网络策略的变化集。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/policies/{policy\_id}/change-set

表 4-345 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
policy_id	是	String	网络策略ID。 最小长度：36 最大长度：36
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

### 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-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

### 响应参数

状态码：200

表 4-347 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
central_network_policy_change_set	Array of CentralNetworkElementChange objects	网络策略变化列表。

表 4-348 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-349 CentralNetworkElementChange

参数	参数类型	描述
operation_id	String	实例状态。 <ul style="list-style-type: none"><li>• CreateCentralNetworkPlane: 新增中心网络平面</li><li>• DeleteCentralNetworkPlane: 移除中心网络平面</li><li>• UpdateCentralNetworkPlane: 更新中心网络平面</li><li>• CreateCentralNetworkErInstance: 新增中心网络ER实例</li><li>• DeleteCentralNetworkErInstance: 移除中心网络ER实例</li><li>• CreateCentralNetworkErConnection: 新增中心网络ER连接</li><li>• DeleteCentralNetworkErConnection: 移除中心网络ER连接</li><li>• CreateCentralNetworkErTable: 新增中心网络ER路由表</li><li>• DeleteCentralNetworkErTable: 移除中心网络ER路由表</li></ul>

## 请求示例

查询中心网络策略变化集

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/policies/{policy_id}/change-set
```

## 响应示例

状态码： 200

查询与当前应用中心网络策略的变化集成功。

```
{
  "request_id": "a334adc3e2cba87855de6f3ab58b9db5",
  "page_info": {
    "current_count": 5
  },
  "central_network_policy_change_set": [ {
    "operation_id": "UpdateCentralNetworkPlane",
    "original_central_network_plane": {
      "name": "default-plane",
      "is_default": true,
      "associate_er_tables": [ {
        "project_id": "XXX",
        "region_id": "cn-north-4",
        "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
        "enterprise_router_table_id": "cc542128-5c2d-402a-8960-53bb2ed9484e"
      } ]
    }
  },
  "newest_central_network_plane": {
```

```
"name": "default-plane",
"is_default": true,
"associate_er_tables": [ {
  "project_id": "XXX",
  "region_id": "cn-north-4",
  "enterprise_router_id": "c73b26b7-33f0-438d-b440-8e87dfe6fef9",
  "enterprise_router_table_id": "c0d51f20-0313-40f7-a74e-9dccb5da21c0"
} ]
}, {
  "operation_id": "CreateCentralNetworkErInstance",
  "create_central_network_er_instance": {
    "enterprise_router_id": "c73b26b7-33f0-438d-b440-8e87dfe6fef9",
    "project_id": "XXX",
    "region_id": "cn-north-4"
  }
}, {
  "operation_id": "DeleteCentralNetworkErInstance",
  "delete_central_network_er_instance": {
    "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
    "project_id": "XXX",
    "region_id": "cn-north-4"
  }
}, {
  "operation_id": "CreateCentralNetworkErTable",
  "central_network_plane_name": "default-plane",
  "project_id": "XXX",
  "region_id": "cn-north-4",
  "enterprise_router_id": "c73b26b7-33f0-438d-b440-8e87dfe6fef9",
  "enterprise_router_table_id": "c0d51f20-0313-40f7-a74e-9dccb5da21c0"
}, {
  "operation_id": "DeleteCentralNetworkErTable",
  "central_network_plane_name": "default-plane",
  "project_id": "XXX",
  "region_id": "cn-north-4",
  "enterprise_router_id": "395b0884-aab4-4bf0-8cb8-7f2da26708dd",
  "enterprise_router_table_id": "cc542128-5c2d-402a-8960-53bb2ed9484e"
} ]
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ListCentralNetworkPolicyChangeSetSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListCentralNetworkPolicyChangeSetRequest request = new
ListCentralNetworkPolicyChangeSetRequest();
try {
    ListCentralNetworkPolicyChangeSetResponse response =
client.listCentralNetworkPolicyChangeSet(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

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

## Go

```
package main

import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询与当前应用中心网络策略的变化集成功。

## 错误码

请参见[错误码](#)。

## 4.8 中心网络的附件管理

### 4.8.1 创建中心网络 GDGW 附件

#### 功能介绍

创建中心网络的GDGW附件。

## 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/gdgw-attachments

表 4-350 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

## 请求参数

表 4-351 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-352 请求 Body 参数

参数	是否必选	参数类型	描述
central_network_gdgw_attachment	是	CreateCentralNetworkGdgwAttachment object	创建中心网络GDGW附件的请求体。

表 4-353 CreateCentralNetworkGdgwAttachment

参数	是否必选	参数类型	描述
name	是	String	实例名字。



参数	是否必选	参数类型	描述
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
enterprise_router_id	是	String	企业路由器的ID。
global_dc_gateway_id	是	String	资源ID标识符。
enterprise_router_project_id	是	String	企业路由器的项目ID。
enterprise_router_region_id	是	String	ER路由器的region ID。
global_dc_gateway_project_id	是	String	实例所属项目ID。
global_dc_gateway_region_id	是	String	RegionID。
central_network_plane_id	否	String	资源ID标识符。

## 响应参数

状态码：202

表 4-354 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network_gdgw_attachment	<b>CentralNetworkGdgwAttachment</b> object	中心GDGW附件详情。

表 4-355 CentralNetworkGdgwAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>

参数	参数类型	描述
bandwidth_size	Integer	带宽值定义，单位Mbps。
is_frozen	Boolean	是否冻结
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。
enterprise_router_attachment_id	String	企业路由器的连接ID。
global_dc_gateway_peer_link_id	String	资源ID标识符。
global_dc_gateway_id	String	资源ID标识符。
global_dc_gateway_project_id	String	实例所属项目ID。
global_dc_gateway_region_id	String	RegionID。
enterprise_router_site_code	String	站点编码定义
global_dc_gateway_site_code	String	站点编码定义

## 请求示例

### 创建中心网络GDGW附件

```
POST /v3/{domain_id}/gcn/central-network/{central_network_id}/gdgw-attachments
```

```
{
  "central_network_gdgw_attachment": {
    "name": "name",
    "description": "description",
    "enterprise_router_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_dc_gateway_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_project_id": "32f7402e18154ce5861c9989df956394",
    "enterprise_router_region_id": "cn-north-4",
    "global_dc_gateway_project_id": "32f7402e18154ce5861c9989df956394",
    "global_dc_gateway_region_id": "cn-north-4",
  }
}
```

```
"central_network_plane_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b"  
}  
}
```

## 响应示例

状态码： 202

接受创建中心网络的GDGW附件的请求。

```
{  
  "request_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
  "central_network_gdgv_attachment" : {  
    "id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "name" : "name",  
    "description" : "description",  
    "domain_id" : "XXX",  
    "state" : "AVAILABLE",  
    "created_at" : "2023-10-09T10:09:28.908Z",  
    "updated_at" : "2023-10-09T10:09:28.908Z",  
    "central_network_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "central_network_plane_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "global_connection_bandwidth_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "bandwidth_type" : "BandwidthPackage",  
    "bandwidth_size" : 20,  
    "is_frozen" : false,  
    "enterprise_router_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "enterprise_router_project_id" : "XXX",  
    "enterprise_router_region_id" : "cn-north-4",  
    "enterprise_router_attachment_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "global_dc_gateway_peer_link_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "global_dc_gateway_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "global_dc_gateway_project_id" : "XXX",  
    "global_dc_gateway_region_id" : "cn-north-4",  
    "enterprise_router_site_code" : "cn-north-bj4",  
    "global_dc_gateway_site_code" : "cn-north-bj4"  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建中心网络GDGW附件

```
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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
public class CreateCentralNetworkGdgvAttachmentSolution {  
  
  public static void main(String[] args) {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    // environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running
```

```
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
CreateCentralNetworkGdgwAttachmentRequest request = new
CreateCentralNetworkGdgwAttachmentRequest();
CreateCentralNetworkGdgwAttachmentRequestBody body = new
CreateCentralNetworkGdgwAttachmentRequestBody();
CreateCentralNetworkGdgwAttachment centralNetworkGdgwAttachmentbody = new
CreateCentralNetworkGdgwAttachment();
centralNetworkGdgwAttachmentbody.withGlobalDcGatewayRegionId("cn-north-4")
    .withCentralNetworkPlaneld("a3bad420-33b8-4e26-9e9b-bdf67aa8e72b")
    .withEnterpriseRouterId("a3bad420-33b8-4e26-9e9b-bdf67aa8e72b")
    .withEnterpriseRouterProjectId("32f7402e18154ce5861c9989df956394")
    .withName("name")
    .withGlobalDcGatewayId("a3bad420-33b8-4e26-9e9b-bdf67aa8e72b")
    .withEnterpriseRouterRegionId("cn-north-4")
    .withDescription("description")
    .withGlobalDcGatewayProjectId("32f7402e18154ce5861c9989df956394");
body.withCentralNetworkGdgwAttachment(centralNetworkGdgwAttachmentbody);
request.withBody(body);
try {
    CreateCentralNetworkGdgwAttachmentResponse response =
client.createCentralNetworkGdgwAttachment(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 创建中心网络GDGW附件

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateCentralNetworkGdgwAttachmentRequest()
    centralNetworkGdgwAttachmentbody = CreateCentralNetworkGdgwAttachment(
        global_dc_gateway_region_id="cn-north-4",
        central_network_plane_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
        enterprise_router_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
        enterprise_router_project_id="32f7402e18154ce5861c9989df956394",
        name="name",
        global_dc_gateway_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
        enterprise_router_region_id="cn-north-4",
        description="description",
        global_dc_gateway_project_id="32f7402e18154ce5861c9989df956394"
    )
    request.body = CreateCentralNetworkGdgwAttachmentRequestBody(
        central_network_gdgw_attachment=centralNetworkGdgwAttachmentbody
    )
    response = client.create_central_network_gdgw_attachment(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 创建中心网络GDGW附件

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateCentralNetworkGdgwAttachmentRequest{}
    centralNetworkPlaneIdCentralNetworkGdgwAttachment:= "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b"
    descriptionCentralNetworkGdgwAttachment:= "description"
    centralNetworkGdgwAttachmentbody := &model.CreateCentralNetworkGdgwAttachment{
        GlobalDcGatewayRegionId: "cn-north-4",
        CentralNetworkPlaneId: &centralNetworkPlaneIdCentralNetworkGdgwAttachment,
```

```
EnterpriseRouterId: "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
EnterpriseRouterProjectId: "32f7402e18154ce5861c9989df956394",
Name: "name",
GlobalDcGatewayId: "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
EnterpriseRouterRegionId: "cn-north-4",
Description: &descriptionCentralNetworkGdgwAttachment,
GlobalDcGatewayProjectId: "32f7402e18154ce5861c9989df956394",
}
request.Body = &model.CreateCentralNetworkGdgwAttachmentRequestBody{
    CentralNetworkGdgwAttachment: centralNetworkGdgwAttachmentbody,
}
response, err := client.CreateCentralNetworkGdgwAttachment(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
202	接受创建中心网络的GDGW附件的请求。

## 错误码

请参见[错误码](#)。

## 4.8.2 查询中心网络 GDGW 附件列表

### 功能介绍

查询中心网络GDGW附件列表。分页查询使用的参数为marker、limit。limit默认值为0，没有指定marker时返回第一条数据。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/gdgw-attachments

表 4-356 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

表 4-357 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围：1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
sort_key	否	String	排序字段。 缺省值：updated_at 最小长度：1 最大长度：36
sort_dir	否	Object	指定排序是升序还是降序(asc为升序，desc为降序)。
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20



参数	是否必选	参数类型	描述
state	否	Array	根据状态查询，可查询多个状态。 数组长度：1 - 20
global_dc_gateway_id	否	Array	根据GDW实例ID过滤列表。 数组长度：1 - 10

## 请求参数

表 4-358 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-359 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
central_network_gdgw_attachments	Array of CentralNetworkGdgwAttachment objects	中心网络GDGW附件列表。 数组长度：0 - 2000

表 4-360 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-361 CentralNetworkGdgwAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>● AVAILABLE (可用)</li><li>● CREATING (创建中)</li><li>● UPDATING (更新中)</li><li>● DELETING (删除中)</li><li>● FREEZING (冻结中)</li><li>● UNFREEZING (解冻中)</li><li>● RECOVERING (恢复中)</li><li>● FAILED (失败)</li><li>● DELETED (已删除)</li><li>● APPROVING (审批中)</li><li>● APPROVED (已审批)</li><li>● UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。

参数	参数类型	描述
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
is_frozen	Boolean	是否冻结
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。
enterprise_router_attachment_id	String	企业路由器的连接ID。
global_dc_gateway_peer_link_id	String	资源ID标识符。
global_dc_gateway_id	String	资源ID标识符。
global_dc_gateway_project_id	String	实例所属项目ID。
global_dc_gateway_region_id	String	RegionID。
enterprise_router_site_code	String	站点编码定义

参数	参数类型	描述
global_dc_gateway_site_code	String	站点编码定义

## 请求示例

查询中心网络附件列表

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/gdgw-attachments
```

## 响应示例

**状态码： 200**

查询中心网络GDGW附件列表成功。

```
{
  "request_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "page_info": {
    "next_marker": "8EDF12u71N...",
    "previous_marker": "8EDF12u71N...",
    "current_count": 1
  },
  "central_network_gdgw_attachments": [ {
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name": "name",
    "description": "description",
    "domain_id": "XXX",
    "state": "AVAILABLE",
    "created_at": "2023-10-09T10:11:11.222Z",
    "updated_at": "2023-10-09T10:11:11.222Z",
    "central_network_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "central_network_plane_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_connection_bandwidth_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "bandwidth_type": "BandwidthPackage",
    "bandwidth_size": 20,
    "is_frozen": false,
    "enterprise_router_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_project_id": "XXX",
    "enterprise_router_region_id": "cn-north-4",
    "enterprise_router_attachment_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_dc_gateway_peer_link_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_dc_gateway_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_dc_gateway_project_id": "XXX",
    "global_dc_gateway_region_id": "cn-north-4",
    "enterprise_router_site_code": "cn-north-bj4",
    "global_dc_gateway_site_code": "cn-north-bj4"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListCentralNetworkGdgwAttachmentsSolution {

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

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

        ListCentralNetworkGdgwAttachmentsRequest request = new
        ListCentralNetworkGdgwAttachmentsRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withSortKey("<sort_key>");
        request.withSortDir();
        request.withId();
        request.withName();
        request.withState();
        request.withGlobalDcGatewayId();
        try {
            ListCentralNetworkGdgwAttachmentsResponse response =
            client.listCentralNetworkGdgwAttachments(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

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

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

# In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListCentralNetworkGdgwAttachmentsRequest()
    request.limit = <limit>
    request.marker = "<marker>"
    request.sort_key = "<sort_key>"
    request.sort_dir =
    request.id =
    request.name =
    request.state =
    request.global_dc_gateway_id =
    response = client.list_central_network_gdgw_attachments(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCentralNetworkGdgwAttachmentsRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
```

```
sortKeyRequest:= "<sort_key>"
request.SortKey = &sortKeyRequest
response, err := client.ListCentralNetworkGdgdwAttachments(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	查询中心网络GDGW附件列表成功。

## 错误码

请参见[错误码](#)。

## 4.8.3 创建中心网络 ER 路由表附件

### 功能介绍

创建中心网络的路由表附件。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/er-route-table-attachments

表 4-362 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 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-Auth-Token	否	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>2000000</b>

表 4-364 请求 Body 参数

参数	是否必选	参数类型	描述
central_network_er_route_table_attachment	是	CreateCentralNetworkErRouteTableAttachment object	创建中心网络ER附件的请求体。

表 4-365 CreateCentralNetworkErRouteTableAttachment

参数	是否必选	参数类型	描述
name	是	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度： <b>0</b> 最大长度： <b>255</b>
enterprise_router_id	是	String	企业路由器的ID。
enterprise_router_project_id	是	String	企业路由器的项目ID。
enterprise_router_region_id	是	String	ER路由器的region ID。
attachment_id	否	String	资源ID标识符。
enterprise_router_table_id	是	String	资源ID标识符。
attached_er_table_project_id	是	String	实例所属项目ID。



参数	是否必选	参数类型	描述
attached_er_table_region_id	是	String	RegionID。
attached_er_id	是	String	资源ID标识符。
attached_er_table_id	是	String	资源ID标识符。
central_network_plane_id	是	String	资源ID标识符。
hosted_cloud	是	String	<ul style="list-style-type: none"> <li>• HWCloud (华为云)</li> <li>• Ireland (爱尔兰)</li> </ul> 缺省值: <b>HWCloud</b>

## 响应参数

状态码： 202

表 4-366 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
central_network_er_route_table_attachment	<b>CentralNetworkErRouteTableAttachment</b> object	企业路由器的路由表附件详情。

表 4-367 CentralNetworkErRouteTableAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
domain_id	String	实例所属账号ID。

参数	参数类型	描述
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
is_frozen	Boolean	是否冻结
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。

参数	参数类型	描述
enterprise_router_attachment_id	String	企业路由器的连接ID。
enterprise_router_table_id	String	资源ID标识符。
enterprise_router_site_code	String	站点编码定义
attached_er_table_id	String	资源ID标识符。
attached_er_table_region_id	String	RegionID。
attached_er_table_project_id	String	实例所属项目ID。
attached_er_table_site_code	String	站点编码定义
attached_er_id	String	资源ID标识符。
attached_er_attachment_id	String	资源ID标识符。
approved_state	String	审批状态。 <ul style="list-style-type: none"><li>● APPROVING (审批中)</li><li>● APPROVED (已审批)</li><li>● UNAPPROVED (审批未通过)</li></ul>
hosted_cloud	String	<ul style="list-style-type: none"><li>● HWCloud (华为云)</li><li>● Ireland (爱尔兰)</li></ul>
reason	String	审批拒绝创建企业路由表附件的原因。 最小长度： <b>0</b> 最大长度： <b>1024</b>

## 请求示例

创建中心网络ER路由表附件

```
POST /v3/{domain_id}/gcn/central-network/{central_network_id}/er-route-table-attachments
```

```
{  
  "central_network_er_route_table_attachment" : {  
    "name" : "name",
```

```
"description" : "description",
"enterprise_router_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"enterprise_router_project_id" : "32f7402e18154ce5861c9989df956394",
"enterprise_router_region_id" : "cn-north-4",
"enterprise_router_table_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"attached_er_table_project_id" : "32f7402e18154ce5861c9989df956394",
"attached_er_table_region_id" : "cn-north-4",
"attached_er_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"attached_er_table_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"central_network_plane_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"hosted_cloud" : "HWCloud"
}
}
```

## 响应示例

**状态码： 202**

接受创建中心网络的路由表附件的请求。

```
{
  "request_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "central_network_er_route_table_attachment" : {
    "id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name" : "name",
    "description" : "description",
    "domain_id" : "XXX",
    "state" : "AVAILABLE",
    "created_at" : "2023-10-09T10:09:28.908Z",
    "updated_at" : "2023-10-09T10:09:28.908Z",
    "central_network_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "central_network_plane_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_connection_bandwidth_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "bandwidth_type" : "BandwidthPackage",
    "bandwidth_size" : 20,
    "is_frozen" : false,
    "enterprise_router_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_project_id" : "XXX",
    "enterprise_router_region_id" : "cn-north-4",
    "enterprise_router_attachment_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_table_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_site_code" : "cn-north-bj4",
    "attached_er_table_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "attached_er_table_region_id" : "cn-north-4",
    "attached_er_table_project_id" : "XXX",
    "attached_er_table_site_code" : "cn-north-bj4",
    "attached_er_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "attached_er_attachment_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "approved_state" : "APPROVING",
    "hosted_cloud" : "HWCloud",
    "reason" : "APPROVING"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建中心网络ER路由表附件

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class CreateCentralNetworkErRouteTableAttachmentSolution {

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

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

        CreateCentralNetworkErRouteTableAttachmentRequest request = new
        CreateCentralNetworkErRouteTableAttachmentRequest();
        CreateCentralNetworkErRouteTableAttachmentsRequestBody body = new
        CreateCentralNetworkErRouteTableAttachmentsRequestBody();
        CreateCentralNetworkErRouteTableAttachment centralNetworkErRouteTableAttachmentbody = new
        CreateCentralNetworkErRouteTableAttachment();

        centralNetworkErRouteTableAttachmentbody.withHostedCloud(CreateCentralNetworkErRouteTableAttachme
        nt.HostedCloudEnum.fromValue("HWCloud"))
            .withEnterpriseRouterId("a3bad420-33b8-4e26-9e9b-bdf67aa8e72b")
            .withEnterpriseRouterProjectId("32f7402e18154ce5861c9989df956394")
            .withEnterpriseRouterRegionId("cn-north-4")
            .withDescription("description")
            .withEnterpriseRouterTableId("a3bad420-33b8-4e26-9e9b-bdf67aa8e72b")
            .withCentralNetworkPlaneId("a3bad420-33b8-4e26-9e9b-bdf67aa8e72b")
            .withName("name")
            .withAttachedErTableRegionId("cn-north-4")
            .withAttachedErTableId("a3bad420-33b8-4e26-9e9b-bdf67aa8e72b")
            .withAttachedErTableProjectId("32f7402e18154ce5861c9989df956394")
            .withAttachedErId("a3bad420-33b8-4e26-9e9b-bdf67aa8e72b");
        body.withCentralNetworkErRouteTableAttachment(centralNetworkErRouteTableAttachmentbody);
        request.withBody(body);
        try {
            CreateCentralNetworkErRouteTableAttachmentResponse response =
            client.createCentralNetworkErRouteTableAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 创建中心网络ER路由表附件

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateCentralNetworkErRouteTableAttachmentRequest()
        centralNetworkErRouteTableAttachmentbody = CreateCentralNetworkErRouteTableAttachment(
            hosted_cloud="HWCloud",
            enterprise_router_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            enterprise_router_project_id="32f7402e18154ce5861c9989df956394",
            enterprise_router_region_id="cn-north-4",
            description="description",
            enterprise_router_table_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            central_network_plane_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            name="name",
            attached_er_table_region_id="cn-north-4",
            attached_er_table_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            attached_er_table_project_id="32f7402e18154ce5861c9989df956394",
            attached_er_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b"
        )
        request.body = CreateCentralNetworkErRouteTableAttachmentsRequestBody(
            central_network_er_route_table_attachment=centralNetworkErRouteTableAttachmentbody
        )
        response = client.create_central_network_er_route_table_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 创建中心网络ER路由表附件

```
package main

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

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateCentralNetworkErRouteTableAttachmentRequest{
        descriptionCentralNetworkErRouteTableAttachment:= "description"
        centralNetworkErRouteTableAttachmentbody := &model.CreateCentralNetworkErRouteTableAttachment{
            HostedCloud:
                model.GetCreateCentralNetworkErRouteTableAttachmentHostedCloudEnum().HW_CLOUD,
            EnterpriseRouterId: "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            EnterpriseRouterProjectId: "32f7402e18154ce5861c9989df956394",
            EnterpriseRouterRegionId: "cn-north-4",
            Description: &descriptionCentralNetworkErRouteTableAttachment,
            EnterpriseRouterTableId: "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            CentralNetworkPlaneld: "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            Name: "name",
            AttachedErTableRegionId: "cn-north-4",
            AttachedErTableId: "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            AttachedErTableProjectId: "32f7402e18154ce5861c9989df956394",
            AttachedErId: "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
        }
    }
    request.Body = &model.CreateCentralNetworkErRouteTableAttachmentsRequestBody{
        CentralNetworkErRouteTableAttachment: centralNetworkErRouteTableAttachmentbody,
    }
    response, err := client.CreateCentralNetworkErRouteTableAttachment(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
202	接受创建中心网络的路由表附件的请求。

## 错误码

请参见[错误码](#)。

## 4.8.4 查询中心网络 ER 路由表附件列表

### 功能介绍

查询中心网络ER路由表附件列表。分页查询使用的参数为marker、limit。limit默认值为0，没有指定marker时返回第一条数据。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/er-route-table-attachments

表 4-368 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

表 4-369 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围：1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096



参数	是否必选	参数类型	描述
sort_key	否	String	排序字段。 缺省值: <b>updated_at</b> 最小长度: <b>1</b> 最大长度: <b>36</b>
sort_dir	否	Object	指定排序是升序还是降序(asc为升序, desc为降序)。
id	否	Array	根据id查询, 可查询多个id。 数组长度: <b>1 - 20</b>
name	否	Array	根据名字查询, 可查询多个名字。 数组长度: <b>1 - 20</b>
state	否	Array	根据状态查询, 可查询多个状态。 数组长度: <b>1 - 20</b>
attachment_instance_id	否	Array	Attachment实例的ID。 数组长度: <b>1 - 10</b>

## 请求参数

表 4-370 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度: <b>0</b> 最大长度: <b>2000000</b>

## 响应参数

状态码: 200

表 4-371 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度: <b>32</b> 最大长度: <b>36</b>

参数	参数类型	描述
page_info	<b>PageInfo</b> object	分页查询页的信息。
central_network_er_route_table_attachments	Array of <b>CentralNetworkErRouteTableAttachment</b> objects	创建路由表附件的返回体 数组长度：0 - 200

表 4-372 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-373 CentralNetworkErRouteTableAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。

参数	参数类型	描述
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
is_frozen	Boolean	是否冻结
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。

参数	参数类型	描述
enterprise_router_attachment_id	String	企业路由器的连接ID。
enterprise_router_table_id	String	资源ID标识符。
enterprise_router_site_code	String	站点编码定义
attached_er_table_id	String	资源ID标识符。
attached_er_table_region_id	String	RegionID。
attached_er_table_project_id	String	实例所属项目ID。
attached_er_table_site_code	String	站点编码定义
attached_er_id	String	资源ID标识符。
attached_er_attachment_id	String	资源ID标识符。
approved_state	String	审批状态。 <ul style="list-style-type: none"><li>● APPROVING (审批中)</li><li>● APPROVED (已审批)</li><li>● UNAPPROVED (审批未通过)</li></ul>
hosted_cloud	String	<ul style="list-style-type: none"><li>● HWCloud (华为云)</li><li>● Ireland (爱尔兰)</li></ul>
reason	String	审批拒绝创建企业路由表附件的原因。 最小长度： <b>0</b> 最大长度： <b>1024</b>

## 请求示例

查询中心网络ER路由表附件列表

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/er-route-table-attachments
```

## 响应示例

**状态码： 200**

查询中心网络ER路由表附件列表成功。

```
{
  "request_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "central_network_er_route_table_attachments": [ {
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name": "name",
    "description": "description",
    "domain_id": "XXX",
    "state": "AVAILABLE",
    "created_at": "2023-10-09T10:09:28.908Z",
    "updated_at": "2023-10-09T10:09:28.908Z",
    "central_network_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "central_network_plane_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_connection_bandwidth_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "bandwidth_type": "BandwidthPackage",
    "bandwidth_size": 20,
    "is_frozen": false,
    "enterprise_router_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_project_id": "XXX",
    "enterprise_router_region_id": "cn-north-4",
    "enterprise_router_attachment_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_table_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_site_code": "cn-north-bj4",
    "attached_er_table_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "attached_er_table_region_id": "cn-north-4",
    "attached_er_table_project_id": "XXX",
    "attached_er_table_site_code": "cn-north-bj4",
    "attached_er_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "attached_er_attachment_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "approved_state": "APPROVING",
    "hosted_cloud": "HWCloud",
    "reason": "APPROVING"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListCentralNetworkErRouteTableAttachmentsSolution {

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

this example, set environment variables CLOUD\_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListCentralNetworkErRouteTableAttachmentsRequest request = new
ListCentralNetworkErRouteTableAttachmentsRequest();
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withSortKey("<sort_key>");
request.withSortDir();
request.withId();
request.withName();
request.withState();
request.withAttachmentInstanceId();
try {
    ListCentralNetworkErRouteTableAttachmentsResponse response =
client.listCentralNetworkErRouteTableAttachments(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCentralNetworkErRouteTableAttachmentsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
```

```
request.sort_key = "<sort_key>"
request.sort_dir =
request.id =
request.name =
request.state =
request.attachment_instance_id =
response = client.list_central_network_er_route_table_attachments(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询中心网络ER路由表附件列表成功。

## 错误码

请参见[错误码](#)。

## 4.8.5 查询中心网络 ER 路由表附件详情

### 功能介绍

查询中心网络ER路由表附件详情。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/er-route-table-attachments/{er\_route\_table\_attachment\_id}

表 4-374 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36
er_route_table_attachment_id	是	String	中心网络ER路由表附件ID。 最小长度：36 最大长度：36



## 请求参数

表 4-375 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-376 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network_er_route_table_attachment	CentralNetworkErRouteTableAttachment object	企业路由器的路由表附件详情。

表 4-377 CentralNetworkErRouteTableAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。

参数	参数类型	描述
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
is_frozen	Boolean	是否冻结
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。

参数	参数类型	描述
enterprise_router_attachment_id	String	企业路由器的连接ID。
enterprise_router_table_id	String	资源ID标识符。
enterprise_router_site_code	String	站点编码定义
attached_er_table_id	String	资源ID标识符。
attached_er_table_region_id	String	RegionID。
attached_er_table_project_id	String	实例所属项目ID。
attached_er_table_site_code	String	站点编码定义
attached_er_id	String	资源ID标识符。
attached_er_attachment_id	String	资源ID标识符。
approved_state	String	审批状态。 <ul style="list-style-type: none"><li>● APPROVING (审批中)</li><li>● APPROVED (已审批)</li><li>● UNAPPROVED (审批未通过)</li></ul>
hosted_cloud	String	<ul style="list-style-type: none"><li>● HWCloud (华为云)</li><li>● Ireland (爱尔兰)</li></ul>
reason	String	审批拒绝创建企业路由表附件的原因。 最小长度： <b>0</b> 最大长度： <b>1024</b>

## 请求示例

查询中心ER网络路由表附件列表

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/er-route-table-attachments/{er_route_table_attachment_id}
```

## 响应示例

**状态码： 200**

查询中心网络ER路由表附件成功。

```
{
  "request_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "central_network_er_route_table_attachment": {
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name": "name",
    "description": "description",
    "domain_id": "XXX",
    "state": "AVAILABLE",
    "created_at": "2023-10-09T10:09:28.908Z",
    "updated_at": "2023-10-09T10:09:28.908Z",
    "central_network_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "central_network_plane_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_connection_bandwidth_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "bandwidth_type": "BandwidthPackage",
    "bandwidth_size": 20,
    "is_frozen": false,
    "enterprise_router_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_project_id": "XXX",
    "enterprise_router_region_id": "cn-north-4",
    "enterprise_router_attachment_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_table_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_site_code": "cn-north-bj4",
    "attached_er_table_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "attached_er_table_region_id": "cn-north-4",
    "attached_er_table_project_id": "XXX",
    "attached_er_table_site_code": "cn-north-bj4",
    "attached_er_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "attached_er_attachment_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "approved_state": "APPROVING",
    "hosted_cloud": "HWCloud",
    "reason": "APPROVING"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ShowCentralNetworkErRouteTableAttachmentSolution {

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

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

ShowCentralNetworkErRouteTableAttachmentRequest request = new
ShowCentralNetworkErRouteTableAttachmentRequest();
try {
    ShowCentralNetworkErRouteTableAttachmentResponse response =
client.showCentralNetworkErRouteTableAttachment(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCentralNetworkErRouteTableAttachmentRequest()
        response = client.show_central_network_er_route_table_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"  
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment 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 := cc.NewCcClient(  
        cc.CcClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ShowCentralNetworkErRouteTableAttachmentRequest{}  
    response, err := client.ShowCentralNetworkErRouteTableAttachment(request)  
    if err == nil {  
        fmt.Printf("%v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	查询中心网络ER路由表附件成功。

## 错误码

请参见[错误码](#)。

## 4.8.6 更新中心网络 ER 路由表附件

### 功能介绍

更新中心网络ER路由表附件。

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/er-route-table-attachments/{er\_route\_table\_attachment\_id}

表 4-378 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36
er_route_table_attachment_id	是	String	中心网络ER路由表附件ID。 最小长度：36 最大长度：36

## 请求参数

表 4-379 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-380 请求 Body 参数

参数	是否必选	参数类型	描述
central_network_er_route_table_attachment	是	<a href="#">UpdateCentralNetworkErRouteTableAttachment</a> object	更新中心网络ER路由表附件的属性详情。

表 4-381 UpdateCentralNetworkErRouteTableAttachment

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255

## 响应参数

状态码：200

表 4-382 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network_er_route_table_attachment	CentralNetworkErRouteTableAttachment object	企业路由器的路由表附件详情。

表 4-383 CentralNetworkErRouteTableAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。



参数	参数类型	描述
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
is_frozen	Boolean	是否冻结
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。

参数	参数类型	描述
enterprise_router_attachment_id	String	企业路由器的连接ID。
enterprise_router_table_id	String	资源ID标识符。
enterprise_router_site_code	String	站点编码定义
attached_er_table_id	String	资源ID标识符。
attached_er_table_region_id	String	RegionID。
attached_er_table_project_id	String	实例所属项目ID。
attached_er_table_site_code	String	站点编码定义
attached_er_id	String	资源ID标识符。
attached_er_attachment_id	String	资源ID标识符。
approved_state	String	审批状态。 <ul style="list-style-type: none"><li>● APPROVING (审批中)</li><li>● APPROVED (已审批)</li><li>● UNAPPROVED (审批未通过)</li></ul>
hosted_cloud	String	<ul style="list-style-type: none"><li>● HWCloud (华为云)</li><li>● Ireland (爱尔兰)</li></ul>
reason	String	审批拒绝创建企业路由表附件的原因。 最小长度： <b>0</b> 最大长度： <b>1024</b>

## 请求示例

更新中心网络ER路由表附件

```
PUT /v3/{domain_id}/gcn/central-network/{central_network_id}/er-route-table-attachments/{er_route_table_attachment_id}
```

```
{  
  "central_network_er_route_table_attachment": {
```

```
"name" : "name",  
"description" : "description"  
}  
}
```

## 响应示例

**状态码： 200**

更新中心网络ER路由表附件成功。

```
{  
  "request_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
  "central_network_er_route_table_attachment" : {  
    "id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "name" : "name",  
    "description" : "description",  
    "domain_id" : "XXX",  
    "state" : "AVAILABLE",  
    "created_at" : "2023-10-09T10:09:28.908Z",  
    "updated_at" : "2023-10-09T10:09:28.908Z",  
    "central_network_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "central_network_plane_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "global_connection_bandwidth_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "bandwidth_type" : "BandwidthPackage",  
    "bandwidth_size" : 20,  
    "is_frozen" : false,  
    "enterprise_router_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "enterprise_router_project_id" : "XXX",  
    "enterprise_router_region_id" : "cn-north-4",  
    "enterprise_router_attachment_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "enterprise_router_table_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "enterprise_router_site_code" : "cn-north-bj4",  
    "attached_er_table_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "attached_er_table_region_id" : "cn-north-4",  
    "attached_er_table_project_id" : "XXX",  
    "attached_er_table_site_code" : "cn-north-bj4",  
    "attached_er_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "attached_er_attachment_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "approved_state" : "APPROVING",  
    "hosted_cloud" : "HWCloud",  
    "reason" : "APPROVING"  
  }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新中心网络ER路由表附件

```
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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
public class UpdateCentralNetworkErRouteTableAttachmentSolution {
```

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

    CcClient client = CcClient.newBuilder()
        .withCredential(auth)
        .withRegion(CcRegion.valueOf("<YOUR REGION>"))
        .build();
    UpdateCentralNetworkErRouteTableAttachmentRequest request = new
UpdateCentralNetworkErRouteTableAttachmentRequest();
    UpdateCentralNetworkErRouteTableAttachmentRequestBody body = new
UpdateCentralNetworkErRouteTableAttachmentRequestBody();
    UpdateCentralNetworkErRouteTableAttachment centralNetworkErRouteTableAttachmentbody = new
UpdateCentralNetworkErRouteTableAttachment();
    centralNetworkErRouteTableAttachmentbody.withName("name")
        .withDescription("description");
    body.withCentralNetworkErRouteTableAttachment(centralNetworkErRouteTableAttachmentbody);
    request.withBody(body);
    try {
        UpdateCentralNetworkErRouteTableAttachmentResponse response =
client.updateCentralNetworkErRouteTableAttachment(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

### 更新中心网络ER路由表附件

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

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

```
.with_credentials(credentials) \  
.with_region(CcRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = UpdateCentralNetworkErRouteTableAttachmentRequest()  
    centralNetworkErRouteTableAttachmentbody = UpdateCentralNetworkErRouteTableAttachment(  
        name="name",  
        description="description"  
    )  
    request.body = UpdateCentralNetworkErRouteTableAttachmentRequestBody(  
        central_network_er_route_table_attachment=centralNetworkErRouteTableAttachmentbody  
    )  
    response = client.update_central_network_er_route_table_attachment(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

## Go

### 更新中心网络ER路由表附件

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"  
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment 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 := cc.NewCcClient(  
        cc.CcClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.UpdateCentralNetworkErRouteTableAttachmentRequest{  
        nameCentralNetworkErRouteTableAttachment:= "name"  
        descriptionCentralNetworkErRouteTableAttachment:= "description"  
        centralNetworkErRouteTableAttachmentbody := &model.UpdateCentralNetworkErRouteTableAttachment{  
            Name: &nameCentralNetworkErRouteTableAttachment,  
            Description: &descriptionCentralNetworkErRouteTableAttachment,  
        }  
    }  
    request.Body = &model.UpdateCentralNetworkErRouteTableAttachmentRequestBody{  
        CentralNetworkErRouteTableAttachment: centralNetworkErRouteTableAttachmentbody,  
    }  
    response, err := client.UpdateCentralNetworkErRouteTableAttachment(request)  
    if err == nil {  
        fmt.Printf("%v\n", response)  
    } else {
```

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

## 更多

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

## 状态码

状态码	描述
200	更新中心网络ER路由表附件成功。

## 错误码

请参见[错误码](#)。

## 4.8.7 查询中心网络 GDGW 附件详情

### 功能介绍

查询中心网络GDGW附件详情。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/gdgw-attachments/{gdgw\_attachment\_id}

表 4-384 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36
gdgw_attachment_id	是	String	中心网络DGW附件ID。 最小长度：36 最大长度：36

## 请求参数

表 4-385 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-386 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network_gdgw_attachment	CentralNetworkGdgwAttachment object	中心GDGW附件详情。

表 4-387 CentralNetworkGdgwAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。

参数	参数类型	描述
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
is_frozen	Boolean	是否冻结
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。



参数	参数类型	描述
enterprise_router_attachment_id	String	企业路由器的连接ID。
global_dc_gateway_peer_link_id	String	资源ID标识符。
global_dc_gateway_id	String	资源ID标识符。
global_dc_gateway_project_id	String	实例所属项目ID。
global_dc_gateway_region_id	String	RegionID。
enterprise_router_site_code	String	站点编码定义
global_dc_gateway_site_code	String	站点编码定义

## 请求示例

查询中心网络GDGW附件

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/gdgw-attachments/{gdgw_attachment_id}
```

## 响应示例

状态码： 200

查询中心网络GDGW附件成功。

```
{
  "request_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "central_network_gdgw_attachment": {
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name": "name",
    "description": "description",
    "domain_id": "XXX",
    "state": "AVAILABLE",
    "created_at": "2023-10-09T10:12:28.127Z",
    "updated_at": "2023-10-09T10:12:28.127Z",
    "central_network_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "central_network_plane_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_connection_bandwidth_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "bandwidth_type": "BandwidthPackage",
    "bandwidth_size": 20,
    "is_frozen": false,
    "enterprise_router_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_project_id": "XXX",
    "enterprise_router_region_id": "cn-north-4",
```

```
"enterprise_router_attachment_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"global_dc_gateway_peer_link_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"global_dc_gateway_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"global_dc_gateway_project_id" : "XXX",
"global_dc_gateway_region_id" : "cn-north-4",
"enterprise_router_site_code" : "cn-north-bj4",
"global_dc_gateway_site_code" : "cn-north-bj4"
}
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.v3.region.CcRegion;
import com.huaweicloud.sdk.v3.*;
import com.huaweicloud.sdk.v3.model.*;

public class ShowCentralNetworkGdgwAttachmentSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowCentralNetworkGdgwAttachmentRequest request = new
        ShowCentralNetworkGdgwAttachmentRequest();
        try {
            ShowCentralNetworkGdgwAttachmentResponse response =
            client.showCentralNetworkGdgwAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCentralNetworkGdgwAttachmentRequest()
        response = client.show_central_network_gdgw_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowCentralNetworkGdgwAttachmentRequest{}
    response, err := client.ShowCentralNetworkGdgwAttachment(request)
```

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

## 更多

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

## 状态码

状态码	描述
200	查询中心网络GDGW附件成功。

## 错误码

请参见[错误码](#)。

## 4.8.8 更新中心网络 GDGW 附件

### 功能介绍

更新中心网络GDGW附件。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/gdgw-attachments/{gdgw\_attachment\_id}

表 4-388 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

参数	是否必选	参数类型	描述
gdgw_attachment_id	是	String	中心网络DGW附件ID。 最小长度：36 最大长度：36

## 请求参数

表 4-389 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-390 请求 Body 参数

参数	是否必选	参数类型	描述
central_network_gdgw_attachment	是	<a href="#">UpdateCentralNetworkGdgwAttachment</a> object	更新中心网络GDGW附件的属性详情。

表 4-391 UpdateCentralNetworkGdgwAttachment

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255

## 响应参数

状态码：200

表 4-392 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network_gdgw_attachment	CentralNetworkGdgwAttachment object	中心GDGW附件详情。

表 4-393 CentralNetworkGdgwAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>● AVAILABLE (可用)</li><li>● CREATING (创建中)</li><li>● UPDATING (更新中)</li><li>● DELETING (删除中)</li><li>● FREEZING (冻结中)</li><li>● UNFREEZING (解冻中)</li><li>● RECOVERING (恢复中)</li><li>● FAILED (失败)</li><li>● DELETED (已删除)</li><li>● APPROVING (审批中)</li><li>● APPROVED (已审批)</li><li>● UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。

参数	参数类型	描述
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
is_frozen	Boolean	是否冻结
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。
enterprise_router_attachment_id	String	企业路由器的连接ID。
global_dc_gateway_peer_link_id	String	资源ID标识符。
global_dc_gateway_id	String	资源ID标识符。
global_dc_gateway_project_id	String	实例所属项目ID。
global_dc_gateway_region_id	String	RegionID。
enterprise_router_site_code	String	站点编码定义

参数	参数类型	描述
global_dc_gateway_site_code	String	站点编码定义

## 请求示例

更新中心网络GDGW附件

```
PUT /v3/{domain_id}/gcn/central-network/{central_network_id}/gdgw-attachments/{gdgw_attachment_id}
{
  "central_network_gdgw_attachment": {
    "name": "name",
    "description": "description"
  }
}
```

## 响应示例

状态码： 200

更新中心网络GDGW附件成功。

```
{
  "request_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "central_network_gdgw_attachment": {
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name": "name",
    "description": "description",
    "domain_id": "XXX",
    "state": "AVAILABLE",
    "created_at": "2023-10-09T10:13:18.045Z",
    "updated_at": "2023-10-09T10:13:18.045Z",
    "central_network_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "central_network_plane_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_connection_bandwidth_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "bandwidth_type": "BandwidthPackage",
    "bandwidth_size": 20,
    "is_frozen": false,
    "enterprise_router_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "enterprise_router_project_id": "XXX",
    "enterprise_router_region_id": "cn-north-4",
    "enterprise_router_attachment_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_dc_gateway_peer_link_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_dc_gateway_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "global_dc_gateway_project_id": "XXX",
    "global_dc_gateway_region_id": "cn-north-4",
    "enterprise_router_site_code": "cn-north-bj4",
    "global_dc_gateway_site_code": "cn-north-bj4"
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新中心网络GDGW附件



```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class UpdateCentralNetworkGdgwAttachmentSolution {

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

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

        UpdateCentralNetworkGdgwAttachmentRequest request = new
        UpdateCentralNetworkGdgwAttachmentRequest();
        UpdateCentralNetworkGdgwAttachmentRequestBody body = new
        UpdateCentralNetworkGdgwAttachmentRequestBody();
        UpdateCentralNetworkGdgwAttachment centralNetworkGdgwAttachmentbody = new
        UpdateCentralNetworkGdgwAttachment();
        centralNetworkGdgwAttachmentbody.setName("name")
            .withDescription("description");
        body.withCentralNetworkGdgwAttachment(centralNetworkGdgwAttachmentbody);
        request.withBody(body);
        try {
            UpdateCentralNetworkGdgwAttachmentResponse response =
            client.updateCentralNetworkGdgwAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 更新中心网络GDGW附件

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateCentralNetworkGdgwAttachmentRequest()
        centralNetworkGdgwAttachmentbody = UpdateCentralNetworkGdgwAttachment(
            name="name",
            description="description"
        )
        request.body = UpdateCentralNetworkGdgwAttachmentRequestBody(
            central_network_gdgw_attachment=centralNetworkGdgwAttachmentbody
        )
        response = client.update_central_network_gdgw_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 更新中心网络GDGW附件

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateCentralNetworkGdgwAttachmentRequest{}
```

```
nameCentralNetworkGdgwAttachment:= "name"
descriptionCentralNetworkGdgwAttachment:= "description"
centralNetworkGdgwAttachmentbody := &model.UpdateCentralNetworkGdgwAttachment{
    Name: &nameCentralNetworkGdgwAttachment,
    Description: &descriptionCentralNetworkGdgwAttachment,
}
request.Body = &model.UpdateCentralNetworkGdgwAttachmentRequestBody{
    CentralNetworkGdgwAttachment: centralNetworkGdgwAttachmentbody,
}
response, err := client.UpdateCentralNetworkGdgwAttachment(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	更新中心网络GDGW附件成功。

## 错误码

请参见[错误码](#)。

## 4.8.9 查询中心网络附件列表

### 功能介绍

查询中心网络附件列表，分页查询使用的参数为marker、limit。limit默认值为0，没有指定marker时返回第一条数据。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/attachments

表 4-394 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

参数	是否必选	参数类型	描述
central_network_id	是	String	中心网络的ID。 最小长度： <b>36</b> 最大长度： <b>36</b>

表 4-395 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值： <b>1</b> 最大值： <b>1000</b> 缺省值： <b>1000</b>
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度： <b>1</b> 最大长度： <b>4096</b>
sort_key	否	String	排序字段。 缺省值： <b>updated_at</b> 最小长度： <b>1</b> 最大长度： <b>36</b>
sort_dir	否	Object	指定排序是升序还是降序(asc为升序，desc为降序)。
id	否	Array	根据id查询，可查询多个id。 数组长度： <b>1 - 20</b>
name	否	Array	根据名字查询，可查询多个名字。 数组长度： <b>1 - 20</b>
attachment_instance_type	否	Array	根据附件类型查询，可查询多个附件类型。 数组长度： <b>1 - 20</b>
state	否	Array	根据状态查询，可查询多个状态。 数组长度： <b>1 - 20</b>

参数	是否必选	参数类型	描述
attachment_instance_id	否	Array	Attachment实例的ID。 数组长度：1 - 10

## 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	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-397 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
central_network_attachments	Array of CentralNetworkAttachment objects	接入网络实例连接列表。 数组长度：0 - 2000

表 4-398 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096

参数	参数类型	描述
previous_marker	String	向前分页标识符。 最小长度： <b>1</b> 最大长度： <b>4096</b>
current_count	Integer	当前列表中资源数量。 最小值： <b>0</b> 最大值： <b>2000</b>

表 4-399 CentralNetworkAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： <b>0</b> 最大长度： <b>255</b>
domain_id	String	实例所属账号ID。
state	String	中心网络附件状态。 <ul style="list-style-type: none"><li>● AVAILABLE (可用)</li><li>● CREATING (创建中)</li><li>● UPDATING (更新中)</li><li>● DELETING (删除中)</li><li>● FREEZING (冻结中)</li><li>● UNFREEZING (解冻中)</li><li>● RECOVERING (恢复中)</li><li>● FAILED (失败)</li><li>● DELETED (已删除)</li><li>● APPROVING (审批中)</li><li>● APPROVED (已审批)</li><li>● UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。

参数	参数类型	描述
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
is_frozen	Boolean	是否冻结
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。
enterprise_router_attachment_id	String	企业路由器的连接ID。
attachment_instance_type	String	接入网络实例类型，GDGW (专线)和ER_ROUTE_TABLE (路由表)。
attachment_instance_id	String	中心网络附件对端实例ID，企业路由器的路由表ID或者GDGW的ID。
attachment_id	String	中心网络附件对端实例的连接ID，企业路由器的连接ID或者GDGW的连接ID。
attachment_instance_project_id	String	中心网络附件对端实例的项目ID。
attachment_instance_region_id	String	中心网络附件对端实例的regionID。
attachment_instance_site_code	String	中心网络附件对端实例的站点编码。
enterprise_router_site_code	String	企业路由器的站点编码。

参数	参数类型	描述
specification_value	CentralNetworkAttachmentSpecificationValueInfo object	附件的额外信息。

表 4-400 CentralNetworkAttachmentSpecificationValueInfo

参数	参数类型	描述
enterprise_router_table_id	String	企业路由器的路由表ID。
attached_enterprise_router_id	String	路由表附件的额外信息，附件的企业路由器ID。
approved_state	String	审批状态。 <ul style="list-style-type: none"><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
hosted_cloud	String	<ul style="list-style-type: none"><li>• HWCloud (华为云)</li><li>• Ireland (爱尔兰)</li></ul>
reason	String	审批拒绝创建附件的原因。 最小长度：0 最大长度：1024

## 请求示例

查询中心网络附件列表

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/attachments
```

## 响应示例

状态码：200

查询接入网络实例连接列表成功。

```
{
  "request_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "page_info": {
    "next_marker": "8EDF12u71N...",
    "previous_marker": "8EDF12u71N...",
    "current_count": 1
  },
  "central_network_attachments": [ {
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name": "name",
    "description": "description",
```



```
"domain_id": "XXX",
"state": "AVAILABLE",
"created_at": "2023-10-09T10:13:57.962Z",
"updated_at": "2023-10-09T10:13:57.962Z",
"central_network_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"central_network_plane_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"global_connection_bandwidth_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"bandwidth_type": "BandwidthPackage",
"bandwidth_size": 20,
"is_frozen": false,
"enterprise_router_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"enterprise_router_project_id": "XXX",
"enterprise_router_region_id": "cn-north-4",
"enterprise_router_attachment_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"attachment_instance_type": "GDGW",
"attachment_instance_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"attachment_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"attachment_instance_project_id": "XXX",
"attachment_instance_region_id": "cn-north-4",
"attachment_instance_site_code": "cn-north-bj4",
"enterprise_router_site_code": "cn-north-bj4",
"specification_value": {
  "enterprise_router_table_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "attached_er_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "approved_state": "APPROVING",
  "hosted_cloud": "HWCloud",
  "reason": "string"
}
}
}
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListCentralNetworkAttachmentsSolution {

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

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

```
        .withRegion(CcRegion.valueOf("<YOUR REGION>"))
        .build();
ListCentralNetworkAttachmentsRequest request = new ListCentralNetworkAttachmentsRequest();
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withSortKey("<sort_key>");
request.withSortDir();
request.withId();
request.withName();
request.withAttachmentInstanceType();
request.withState();
request.withAttachmentInstanceId();
try {
    ListCentralNetworkAttachmentsResponse response = client.listCentralNetworkAttachments(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCentralNetworkAttachmentsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.sort_key = "<sort_key>"
        request.sort_dir =
        request.id =
        request.name =
        request.attachment_instance_type =
        request.state =
        request.attachment_instance_id =
        response = client.list_central_network_attachments(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询接入网络实例连接列表成功。

## 错误码

请参见[错误码](#)。

## 4.8.10 删除中心网络附件

### 功能介绍

删除中心网络附件。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/attachments/{attachment\_id}

表 4-401 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36
attachment_id	是	String	中心网络附件ID。 最小长度：36 最大长度：36

### 请求参数

表 4-402 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

### 响应参数

状态码：202

表 4-403 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
central_network_attachment	CentralNetworkAttachment object	中心网络附件详情。

表 4-404 CentralNetworkAttachment

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
state	String	中心网络附件状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
central_network_id	String	中心网络ID。

参数	参数类型	描述
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
is_frozen	Boolean	是否冻结
enterprise_router_id	String	企业路由器的ID。
enterprise_router_project_id	String	企业路由器的项目ID。
enterprise_router_region_id	String	ER路由器的region ID。
enterprise_router_attachment_id	String	企业路由器的连接ID。
attachment_instance_type	String	接入网络实例类型，GDGW (专线)和ER_ROUTE_TABLE (路由表)。
attachment_instance_id	String	中心网络附件对端实例ID，企业路由器的路由表ID或者GDGW的ID。
attachment_id	String	中心网络附件对端实例的连接ID，企业路由器的连接ID或者GDGW的连接ID。
attachment_instance_project_id	String	中心网络附件对端实例的项目ID。
attachment_instance_region_id	String	中心网络附件对端实例的regionID。
attachment_instance_site_code	String	中心网络附件对端实例的站点编码。
enterprise_router_site_code	String	企业路由器的站点编码。

参数	参数类型	描述
specification_value	CentralNetworkAttachmentSpecificationValueInfo object	附件的额外信息。

表 4-405 CentralNetworkAttachmentSpecificationValueInfo

参数	参数类型	描述
enterprise_router_table_id	String	企业路由器的路由表ID。
attached_enter_id	String	路由表附件的额外信息，附件的企业路由器ID。
approved_state	String	审批状态。 <ul style="list-style-type: none"><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
hosted_cloud	String	<ul style="list-style-type: none"><li>• HWCloud (华为云)</li><li>• Ireland (爱尔兰)</li></ul>
reason	String	审批拒绝创建附件的原因。 最小长度：0 最大长度：1024

## 请求示例

删除中心网络附件

```
DELETE /v3/{domain_id}/gcn/central-network/{central_network_id}/attachments/{attachment_id}
```

## 响应示例

状态码：202

接受删除中心网络附件。

```
{
  "request_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "central_network_attachment" : {
    "id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name" : "name",
    "description" : "description",
    "domain_id" : "XXX",
    "state" : "AVAILABLE",
    "created_at" : "2023-10-09T10:15:32.134Z",
    "updated_at" : "2023-10-09T10:15:32.134Z",
    "central_network_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
```

```
"central_network_plane_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"global_connection_bandwidth_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"bandwidth_type" : "BandwidthPackage",
"bandwidth_size" : 20,
"is_frozen" : false,
"enterprise_router_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"enterprise_router_project_id" : "XXX",
"enterprise_router_region_id" : "cn-north-4",
"enterprise_router_attachment_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"attachment_instance_type" : "GDGW",
"attachment_instance_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"attachment_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
"attachment_instance_project_id" : "XXX",
"attachment_instance_region_id" : "cn-north-4",
"attachment_instance_site_code" : "cn-north-bj4",
"enterprise_router_site_code" : "cn-north-bj4",
"specification_value" : {
  "enterprise_router_table_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "attached_er_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "approved_state" : "APPROVING",
  "hosted_cloud" : "HWCloud",
  "reason" : "string"
}
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteCentralNetworkAttachmentSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteCentralNetworkAttachmentRequest request = new DeleteCentralNetworkAttachmentRequest();
        try {
            DeleteCentralNetworkAttachmentResponse response =
                client.deleteCentralNetworkAttachment(request);
            System.out.println(response.toString());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



```
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteCentralNetworkAttachmentRequest()
        response = client.delete_central_network_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
202	接受删除中心网络附件。

## 错误码

请参见[错误码](#)。

# 4.9 中心网络连接

## 4.9.1 查询中心网络连接列表

### 功能介绍

查询中心网络连接列表接口。分页查询使用的参数为marker、limit。limit默认值为0，没有指定marker时返回第一条数据。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/connections

表 4-406 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36

表 4-407 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围：1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
sort_key	否	String	排序字段。 缺省值：updated_at 最小长度：1 最大长度：36
sort_dir	否	Object	指定排序是升序还是降序(asc为升序，desc为降序)。
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20

参数	是否必选	参数类型	描述
state	否	Array	根据状态查询，可查询多个状态。 数组长度：1 - 20
global_connection_bandwidth_id	否	Array	根据带宽包ID过滤。 数组长度：1 - 20
bandwidth_type	否	Object	根据带宽类型查询。带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
connection_type	否	Object	连接类型，支持。
is_cross_region	否	Boolean	是否跨地域。

## 请求参数

表 4-408 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-409 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36

参数	参数类型	描述
page_info	<b>PageInfo</b> object	分页查询页的信息。
central_network_connections	Array of <b>CentralNetworkConnections</b> objects	中心网络连接列表。 数组长度：0 - 2000

表 4-410 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-411 CentralNetworkConnection

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。

参数	参数类型	描述
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
is_frozen	Boolean	是否冻结
connection_type	String	中心网络连接类型定义： <ul style="list-style-type: none"><li>• ER-ER (ER-ER Connection)</li><li>• ER-GDGW (ER-GDGW Attachment Connection)</li><li>• ER-ER_ROUTE_TABLE (ER-ER_ROUTE_TABLE Attachment Connection)</li></ul>
connection_point_pair	Array of <b>ConnectionPoint</b> objects	中心网络连接的两个端点定义，长度固定为2的数组。 数组长度： <b>2 - 2</b>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。

表 4-412 ConnectionPoint

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
site_code	String	站点编码定义
instance_id	String	资源ID标识符。
type	String	中心网络连接点类型定义： <ul style="list-style-type: none"><li>● ER (EnterpriseRouter)</li><li>● GDGW (Global DC Gateway)</li><li>● ER_ROUTE_TABLE (Route Table)</li></ul>

## 请求示例

查询中心网络连接列表

```
GET /v3/{domain_id}/gcn/central-network/{central_network_id}/connections
```

## 响应示例

状态码： 200

查询中心网络连接列表成功。

```
{
  "request_id": "0050ea65af7e190d9a791d9e69f28e63",
  "page_info": {
    "current_count": 1
  },
  "central_network_connections": [ {
    "id": "eb350a5c-06c1-4b12-9ae4-3820f31faaa0",
    "name": "er-connection",
    "domain_id": "XXX",
    "enterprise_project_id": "0",
    "central_network_id": "43ad756e-2780-463e-a8ed-27a5bd77b7a7",
    "central_network_plane_id": "238f60d5-ed1a-4508-8473-45509c530c8f",
    "bandwidth_type": "TestBandwidth",
    "state": "AVAILABLE",
    "is_frozen": false,
    "connection_type": "ER-ER",
    "connection_point_pair": [ {
      "id": "1c233723-234c-4b50-8fba-7072b4c6aa1a",
      "project_id": "XXX",
      "region_id": "cn-north-4",
      "site_code": "cn-north-4",
      "instance_id": "d238dd85-f769-485b-aa05-d20e1d3889df",
      "type": "ER"
    }
  ], {
    "id": "8641d7f0-bd0c-49fe-8b70-3649cd7dd9b1",
    "project_id": "XXX",
    "region_id": "cn-north-1",
    "site_code": "cn-north-1",
    "instance_id": "96abbb18-4a77-4103-aa8e-b1c836154282",
    "type": "ER"
  }
  ]
}
```

```
    }],  
    "created_at" : "2023-10-09T08:02:43.31Z",  
    "updated_at" : "2023-10-09T08:02:43.307Z"  
  }]  
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ListCentralNetworkConnectionsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CcClient client = CcClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListCentralNetworkConnectionsRequest request = new ListCentralNetworkConnectionsRequest();  
        request.withLimit(<limit>);  
        request.withMarker("<marker>");  
        request.withSortKey("<sort_key>");  
        request.withSortDir();  
        request.withId();  
        request.withName();  
        request.withState();  
        request.withGlobalConnectionBandwidthId();  
        request.withBandwidthType();  
        request.withConnectionType();  
        request.withIsCrossRegion(<is_cross_region>);  
        try {  
            ListCentralNetworkConnectionsResponse response = client.listCentralNetworkConnections(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCentralNetworkConnectionsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.sort_key = "<sort_key>"
        request.sort_dir =
        request.id =
        request.name =
        request.state =
        request.global_connection_bandwidth_id =
        request.bandwidth_type =
        request.connection_type =
        request.is_cross_region = <IsCrossRegion>
        response = client.list_central_network_connections(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

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

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

```
example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListCentralNetworkConnectionsRequest{
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    sortKeyRequest:= "<sort_key>"
    request.SortKey = &sortKeyRequest
    isCrossRegionRequest:= <is_cross_region>
    request.IsCrossRegion = &isCrossRegionRequest
    response, err := client.ListCentralNetworkConnections(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	查询中心网络连接列表成功。

## 错误码

请参见[错误码](#)。

## 4.9.2 更新中心网络连接接口

### 功能介绍

更新中心网络连接接口（仅支持更新带宽）。

### 调用方法

请参见[如何调用API](#)。

## URI

PUT /v3/{domain\_id}/gcn/central-network/{central\_network\_id}/connections/  
{connection\_id}

表 4-413 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
central_network_id	是	String	中心网络的ID。 最小长度：36 最大长度：36
connection_id	是	String	中心网络连接ID。 最小长度：36 最大长度：36

## 请求参数

表 4-414 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-415 请求 Body 参数

参数	是否必选	参数类型	描述
central_network_connection	是	<b>UpdateCentralNetworkConnection</b> object	更新中心网络连接信息。* 带宽类型：按照带宽包分配、按照流量计费（当前版本暂不支持，预留后续支持按流量计费模式）、测试带宽类型 * 带宽包ID：按照带宽包分配必填，其他类型不填；* 带宽取值：按照带宽包分配和按照流量计费必填，代表限速带宽，按照带宽包分配下不能超过带宽包大小，测试带宽类型不填；

表 4-416 UpdateCentralNetworkConnection

参数	是否必选	参数类型	描述
bandwidth_type	是	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
global_connection_bandwidth_id	否	String	全域互联带宽ID。
bandwidth_size	否	Integer	带宽值定义，单位Mbps。

## 响应参数

状态码： 202

表 4-417 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
central_network_connection	<b>CentralNetworkConnection</b> object	中心网络连接。

表 4-418 CentralNetworkConnection

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
domain_id	String	实例所属账号ID。

参数	参数类型	描述
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
central_network_id	String	中心网络ID。
central_network_plane_id	String	中心网络平面ID。
global_connection_bandwidth_id	String	全域互联带宽ID。
bandwidth_type	String	带宽类型包括： <ul style="list-style-type: none"><li>• BandwidthPackage (按带宽计费，需要绑定全域互联带宽，并指定分配带宽大小)</li><li>• TestBandwidth (不收费的测试带宽，仅保留最小带宽，用于测试跨地域连通性)</li></ul>
bandwidth_size	Integer	带宽值定义，单位Mbps。
state	String	中心网络连接状态。 <ul style="list-style-type: none"><li>• AVAILABLE (可用)</li><li>• CREATING (创建中)</li><li>• UPDATING (更新中)</li><li>• DELETING (删除中)</li><li>• FREEZING (冻结中)</li><li>• UNFREEZING (解冻中)</li><li>• RECOVERING (恢复中)</li><li>• FAILED (失败)</li><li>• DELETED (已删除)</li><li>• APPROVING (审批中)</li><li>• APPROVED (已审批)</li><li>• UNAPPROVED (审批未通过)</li></ul>
is_frozen	Boolean	是否冻结
connection_type	String	中心网络连接类型定义： <ul style="list-style-type: none"><li>• ER-ER (ER-ER Connection)</li><li>• ER-GDGW (ER-GDGW Attachment Connection)</li><li>• ER-ER_ROUTE_TABLE (ER-ER_ROUTE_TABLE Attachment Connection)</li></ul>

参数	参数类型	描述
connection_point_pair	Array of <a href="#">ConnectionPoint</a> objects	中心网络连接的两个端点定义，长度固定为2的数组。 数组长度：2 - 2
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。

表 4-419 ConnectionPoint

参数	参数类型	描述
id	String	资源ID标识符。
project_id	String	实例所属项目ID。
region_id	String	RegionID。
site_code	String	站点编码定义
instance_id	String	资源ID标识符。
type	String	中心网络连接点类型定义： <ul style="list-style-type: none"><li>ER (EnterpriseRouter)</li><li>GDGW (Global DC Gateway)</li><li>ER_ROUTE_TABLE (Route Table)</li></ul>

## 请求示例

### 更新中心网络连接

```
PUT /v3/{domain_id}/gcn/central-network/{central_network_id}/connections
{
  "central_network_connection": {
    "bandwidth_type": "BandwidthPackage",
    "global_connection_bandwidth_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "bandwidth_size": 20
  }
}
```

## 响应示例

### 状态码：202

接受更新中心网络连接的请求。

```
{
  "request_id": "e108915cd5de228ef252be95ce6ef2c2",
  "central_network_connection": {
```

```
"id": "eb350a5c-06c1-4b12-9ae4-3820f31faaa0",
"name": "er-connection",
"domain_id": "XXX",
"enterprise_project_id": "0",
"central_network_id": "43ad756e-2780-463e-a8ed-27a5bd77b7a7",
"central_network_plane_id": "238f60d5-ed1a-4508-8473-45509c530c8f",
"bandwidth_type": "TestBandwidth",
"state": "UPDATING",
"is_frozen": false,
"connection_type": "ER-ER",
"connection_point_pair": [ {
  "id": "1c233723-234c-4b50-8fba-7072b4c6aa1a",
  "project_id": "XXX",
  "region_id": "cn-north-4",
  "site_code": "cn-north-4",
  "instance_id": "d238dd85-f769-485b-aa05-d20e1d3889df",
  "type": "ER"
}, {
  "id": "8641d7f0-bd0c-49fe-8b70-3649cd7dd9b1",
  "project_id": "XXX",
  "region_id": "cn-north-1",
  "site_code": "cn-north-1",
  "instance_id": "96abbb18-4a77-4103-aa8e-b1c836154282",
  "type": "ER"
}],
"created_at": "2023-10-09T08:02:43.31Z",
"updated_at": "2023-10-09T08:02:43.307Z"
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

#### 更新中心网络连接

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class UpdateCentralNetworkConnectionSolution {

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

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

```
        .withRegion(CcRegion.valueOf("<YOUR REGION>"))
        .build();
        UpdateCentralNetworkConnectionRequest request = new UpdateCentralNetworkConnectionRequest();
        UpdateCentralNetworkConnectionRequestBody body = new
UpdateCentralNetworkConnectionRequestBody();
        UpdateCentralNetworkConnection centralNetworkConnectionbody = new
UpdateCentralNetworkConnection();
        centralNetworkConnectionbody.withGlobalConnectionBandwidthId("a3bad420-33b8-4e26-9e9b-
bdf67aa8e72b")
        .withBandwidthType(UpdateCentralNetworkConnection.BandwidthTypeEnum.fromValue("Bandwidth
Package"))
        .withBandwidthSize(20L);
        body.withCentralNetworkConnection(centralNetworkConnectionbody);
        request.withBody(body);
        try {
            UpdateCentralNetworkConnectionResponse response =
client.updateCentralNetworkConnection(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 更新中心网络连接

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateCentralNetworkConnectionRequest()
        central_network_connection_body = UpdateCentralNetworkConnection(
            global_connection_bandwidth_id="a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
            bandwidth_type="BandwidthPackage",
            bandwidth_size=20
        )
        request.body = UpdateCentralNetworkConnectionRequestBody(
            central_network_connection=central_network_connection_body
        )
```



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

## Go

### 更新中心网络连接

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateCentralNetworkConnectionRequest{
        globalConnectionBandwidthIdCentralNetworkConnection:= "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b"
        bandwidthTypeCentralNetworkConnection:=
model.GetUpdateCentralNetworkConnectionBandwidthTypeEnum().BANDWIDTH_PACKAGE
        bandwidthSizeCentralNetworkConnection:= int64(20)
        centralNetworkConnectionbody := &model.UpdateCentralNetworkConnection{
            GlobalConnectionBandwidthId: &globalConnectionBandwidthIdCentralNetworkConnection,
            BandwidthType: &bandwidthTypeCentralNetworkConnection,
            BandwidthSize: &bandwidthSizeCentralNetworkConnection,
        }
    }
    request.Body = &model.UpdateCentralNetworkConnectionRequestBody{
        CentralNetworkConnection: centralNetworkConnectionbody,
    }
    response, err := client.UpdateCentralNetworkConnection(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
202	接受更新中心网络连接的请求。

## 错误码

请参见[错误码](#)。

# 4.10 云连接配额

## 4.10.1 查询云连接配额

### 功能介绍

查询云连接配额。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/ccaas/quotas

表 4-420 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-421 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000

参数	是否必选	参数类型	描述
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
quota_type	是	String	配额类型： <ul style="list-style-type: none"><li>cloud_connection: 可加载的云连接实例数</li><li>cloud_connection_region: 某云连接实例下可加载的Region数</li><li>cloud_connection_route: 某云连接实例下可加载的路由数</li><li>region_network_instance: 某云连接实例下某个Region下可加载的网络实例数</li></ul> 最小长度：0 最大长度：36
cloud_connection_id	否	String	云连接ID。当查询cloud_connection_region、cloud_connection_route、region_network_instance三种类型的配额时需要填写此参数。 最小长度：0 最大长度：36
region_id	否	String	区域ID。当查询region_network_instance类型的配额时需要填写此参数。 最小长度：0 最大长度：36

## 请求参数

表 4-422 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-423 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
quotas	Array of <a href="#">CloudConnectionQuota</a> objects	配额列表。 数组长度：0 - 2000

表 4-424 CloudConnectionQuota

参数	参数类型	描述
domain_id	String	实例所属账号ID。
cloud_connection_id	String	资源ID标识符。
region_id	String	RegionID。
quota_type	String	配额类型： <ul style="list-style-type: none"><li>cloud_connection: 可加载的云连接实例数</li><li>cloud_connection_region: 某云连接实例下可加载的Region数</li><li>cloud_connection_route: 某云连接实例下可加载的路由数</li><li>region_network_instance: 某云连接实例下某个Region下可加载的网络实例数</li></ul>

参数	参数类型	描述
quota_number	Integer	配额数量。 最小值：0 最大值：102400
quota_used	Integer	配额使用数量。 最小值：0 最大值：102400

## 请求示例

查询资源配额

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/quotas?quota_type=cloud_connection
```

## 响应示例

状态码：200

查询云连接配额成功。

```
{
  "request_id": "XXX",
  "quotas": [ {
    "domain_id": "XXX",
    "quota_type": "cloud_connection",
    "quota_number": 6,
    "quota_used": 3
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ListCloudConnectionQuotasSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListCloudConnectionQuotasRequest request = new ListCloudConnectionQuotasRequest();

request.withQuotaType(ListCloudConnectionQuotasRequest.QuotaTypeEnum.fromValue("<quota_type>"));
request.withCloudConnectionId("<cloud_connection_id>");
request.withRegionId("<region_id>");
request.withLimit(<limit>);
request.withMarker("<marker>");
try {
    ListCloudConnectionQuotasResponse response = client.listCloudConnectionQuotas(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCloudConnectionQuotasRequest()
        request.quota_type = "<quota_type>"
        request.cloud_connection_id = "<cloud_connection_id>"
        request.region_id = "<region_id>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.list_cloud_connection_quotas(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCloudConnectionQuotasRequest{}
    request.QuotaType = model.GetListCloudConnectionQuotasRequestQuotaTypeEnum().<QUOTA_TYPE>
    cloudConnectionIdRequest := "<cloud_connection_id>"
    request.CloudConnectionId = &cloudConnectionIdRequest
    regionIdRequest := "<region_id>"
    request.RegionId = &regionIdRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListCloudConnectionQuotas(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	查询云连接配额成功。

## 错误码

请参见[错误码](#)。

## 4.11 中心网络配额

### 4.11.1 查询中心网络配额

#### 功能介绍

查询中心网络配额。

#### 调试

您可以在[API Explorer](#)中调试该接口，支持自动认证鉴权。API Explorer可以自动生成SDK代码示例，并提供SDK代码示例调试功能。

#### URI

GET /v3/{domain\_id}/gcn/quotas

表 4-425 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-426 Query 参数

参数	是否必选	参数类型	描述
quota_type	否	Array	根据配额类型查询，可查询多个类型。 数组长度：1 - 20
limit	否	Integer	每页返回的个数。取值范围：1~1000。 最小值：1 最大值：1000 缺省值：1000



参数	是否必选	参数类型	描述
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096

## 请求参数

表 4-427 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-428 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
quotas	Array of <a href="#">CentralNetworkQuota</a> objects	配额列表 数组长度：0 - 2000

表 4-429 CentralNetworkQuota

参数	参数类型	描述
quota_key	String	中心网络配额类型。 - central_networks_per_account (每个账户的中心网数) - policy_versions_per_central_network (每个中心网的策略数) - size_of_document_per_central_network_policy_version (中心网络策略文档大小(KB)) - planes_per_central_network (每个中心网的平面数) - er_instances_per_region_per_central_network (每个中心网络每个区域的Er实例数) - connections_per_central_network (每个中心网的连接数) - attachments_per_central_network (每个中心网的附件数) - GDGW_attachments_per_region_per_central_network (每个中心网每个区域的GDGW附件数) - ER_ROUTE_TABLE_attachments_per_region_per_central_network (每个中心网每个区域的ER_ROUTE_TABLE附件数)
quota_limit	Integer	配额大小。 最小值： <b>0</b> 最大值： <b>2147483647</b>
used	Integer	已使用配额。 最小值： <b>0</b> 最大值： <b>2147483647</b>
unit	String	配额值的单位。 最小长度： <b>0</b> 最大长度： <b>255</b>

## 请求示例

查询租户配额

```
GET https://{cc_endpoint}/v3/{domain_id}/gcn/quotas
```

## 响应示例

状态码：**200**

查询中心网络配额成功

```
{  
  "request_id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
  "quotas": [  
    {  
      "quota_key": "central_networks_per_account",  
      "quota_limit": 6,  
      "used": 0,  
      "unit": "count"  
    }  
  ]  
}
```

```
  }]  
}
```

## 状态码

状态码	描述
200	查询中心网络配额成功

## 错误码

请参见[错误码](#)。

# 4.12 中心网络能力

## 4.12.1 查询中心网络能力列表

### 功能介绍

查询中心网络能力列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcn/capabilities

表 4-430 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-431 Query 参数

参数	是否必选	参数类型	描述
capability	否	Array	根据租户能力名查询，可查询多个类型。 数组长度：1 - 20

## 请求参数

表 4-432 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-433 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
capabilities	Array of <a href="#">CentralNetworkCapability</a> objects	租户能力列表 数组长度：0 - 2000

表 4-434 CentralNetworkCapability

参数	参数类型	描述
id	String	资源ID标识符。
domain_id	String	实例所属账号ID。

参数	参数类型	描述
capability	String	能力类型定义： <ul style="list-style-type: none"><li>• central-network.is-support: 支持中心网络</li><li>• central-network.is-support-enterprise-project: 支持中心网络的企业项目</li><li>• central-network.is-support-tag: 支持中心网络的标签能力</li><li>• central-network.is-support-custom-er-table: 支持中心网络的自定义路由表能力</li><li>• connection-bandwidth.size-range: 跨地域连接带宽大小的范围</li><li>• connection-bandwidth.charge-mode: 跨地域连接带宽计费类型</li><li>• er-instance.support-regions: 支持ER实例的Region列表</li><li>• er-instance.support-ipv6-regions: 支持IPV6的ER实例Region列表</li><li>• er-instance.support-dscp-regions: 支持带宽包金银铜配置的region列表</li><li>• er-instance.support-sts5-regions: 支持sts5调用ER服务的region列表</li><li>• er-instance.support-sites: 支持ER实例的站点列表</li><li>• gdgw-attachment.is-support: 支持GDGW附件</li><li>• gdgw-attachment.support-regions: 支持GDGW附件的Region列表</li><li>• gdgw-attachment.support-sites: 支持GDGW附件的站点列表</li><li>• er-route-table-attachment.is-support: 支持路由表附件</li><li>• er-route-table-attachment.support-regions: 支持路由表附件的Region列表</li><li>• er-route-table-attachment.support-sites: 支持路由表附件的站点列表</li><li>• cloud-alliance.support-regions: 支持云联盟场景的Region列表</li></ul>

## 请求示例

查询租户能力

GET [https://{cc\\_endpoint}/v3/gcn/capabilities](https://{cc_endpoint}/v3/gcn/capabilities)

## 响应示例

**状态码： 200**

查询中心网络能力列表成功。

```
{
  "request_id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "capabilities" : [ {
    "id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "domain_id" : "XXX",
    "capability" : "central-network.is-support",
    "specifications" : {
      "is_support" : true
    }
  }
]
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListCentralNetworkCapabilitiesSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCentralNetworkCapabilitiesRequest request = new ListCentralNetworkCapabilitiesRequest();
        request.withCapability();
        try {
            ListCentralNetworkCapabilitiesResponse response = client.listCentralNetworkCapabilities(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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListCentralNetworkCapabilitiesRequest()
        request.capability =
        response = client.list_central_network_capabilities(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询中心网络能力列表成功。

## 错误码

请参见[错误码](#)。

# 4.13 全域互联带宽

## 4.13.1 查询全域互联带宽列表

### 功能介绍

查询全域互联带宽列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcb/gcbbandwidths



表 4-435 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-436 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
name	否	Array	根据名字查询，可查询多个名字。 数组长度：1 - 20
enterprise_project_id	否	Array	根据企业项目ID过滤列表。 数组长度：1 - 20
instance_id	否	Array	根据绑定实例id过滤全域互联带宽列表。 数组长度：1 - 10

参数	是否必选	参数类型	描述
instance_type	否	Array	根据绑定实例类型过滤全域互联带宽列表。实例类型： <ul style="list-style-type: none"><li>● CC: 云连接</li><li>● GEIP: 全域弹性公网IP</li><li>● GCN: 中心网络</li><li>● GSN: 分支网络</li></ul> 数组长度：1 - 10
binding_service	否	Array	根据支持绑定实例类型过滤全域互联带宽列表。实例类型： <ul style="list-style-type: none"><li>● CC: 云连接</li><li>● GEIP: 全域弹性公网IP</li><li>● GCN: 中心网络</li><li>● GSN: 分支网络</li></ul> 数组长度：1 - 10
type	否	Array	根据带宽类型过滤全域互联带宽列表。带宽类型： <ul style="list-style-type: none"><li>● TrsArea: 跨区带宽</li><li>● Area: 大区带宽</li><li>● SubArea: 区域带宽</li><li>● Region: 城域带宽</li></ul> 数组长度：1 - 10
admin_state	否	Array	根据带宽状态过滤全域互联带宽列表： <ul style="list-style-type: none"><li>● NORMAL: 正常</li><li>● FREEZED: 冻结</li></ul> 数组长度：1 - 10
charge_mode	否	Array	根据计费方式过滤全域互联带宽列表： <ul style="list-style-type: none"><li>● bwd: 按带宽计费</li><li>● 95: 按传统型95计费</li></ul> 数组长度：1 - 10

## 请求参数

表 4-437 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-438 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
globalconnection_bandwidths	Array of GlobalConnectionBandwidth objects	全域互联带宽列表响应体。 数组长度：1 - 2000

表 4-439 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-440 GlobalConnectionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： <b>0</b> 最大长度： <b>255</b>
domain_id	String	实例所属账号ID。
bordercross	Boolean	功能说明：全域互联带宽是否跨境，判断依据：带宽是否涉及从中国大陆到其他国家。取值范围：True：跨境；False：非跨境
type	String	功能说明：描述带宽类型，对应地理区间的城域、区域、大区、跨区四级： <ul style="list-style-type: none"><li>• TrsArea: 跨区带宽</li><li>• Area: 大区带宽</li><li>• SubArea: 区域带宽</li><li>• Region: 城域带宽</li></ul>
binding_service	String	功能说明：绑定的服务类型。实例类型： <ul style="list-style-type: none"><li>• CC: 云连接</li><li>• GEIP: 全域弹性公网IP</li><li>• GCN: 中心网络</li><li>• GSN: 分支网络</li><li>• ALL: 所有实例类型</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度： <b>1</b> 最大长度： <b>36</b>
charge_mode	String	功能说明：描述计费类型，描述可选计费类型。默认开放按带宽计费，传统95计费租户白名单控制。取值范围：bwd: 按带宽计费 95: 按传统型95计费
size	Integer	功能说明：全域互联带宽实例中的带宽值大小，单位Mbit/s。取值范围：2-300Mbit/s 最小值： <b>2</b> 最大值： <b>300</b>

参数	参数类型	描述
sla_level	String	功能说明：描述网络等级，从高到低分为铂金、金、银。默认金，其余租户白名单控制。 <ul style="list-style-type: none"><li>• Pt: 铂金</li><li>• Au: 金</li><li>• Ag: 银</li></ul> 缺省值： <b>Au</b> 最小长度： <b>1</b> 最大长度： <b>36</b>
local_area	String	功能说明：本端接入点的中英文名。通过HEADER里面的x-language控制，默认英文，zh-cn返回中文。 最小长度： <b>1</b> 最大长度： <b>36</b>
remote_area	String	功能说明：远端接入点的中英文名。通过HEADER里面的x-language控制，默认英文，zh-cn返回中文。 最小长度： <b>1</b> 最大长度： <b>36</b>
local_site_code	String	功能说明：本端接入点的编码。 最小长度： <b>1</b> 最大长度： <b>36</b>
remote_site_code	String	功能说明：远端接入点的编码。 最小长度： <b>1</b> 最大长度： <b>36</b>
admin_state	String	功能说明：全域互联带宽状态。取值范围：NORMAL-正常 FREEZED-冻结状态
frozen	Boolean	功能说明：全域互联带宽是否冻结。取值范围：true-冻结 false-非冻结
spec_code_id	String	功能说明：线路规格编码UUID。 最小长度： <b>1</b> 最大长度： <b>36</b>
tags	Array of <b>Tag</b> objects	实例标签。 数组长度： <b>1 - 20</b>
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。

参数	参数类型	描述
enable_share	Boolean	功能说明: 全域互联带宽是否支持绑定多实例。 取值范围: true-支持 false-不支持
instances	Array of <b>GlobalConnectionBandwidthAssociateInstance</b> objects	功能说明: 全域互联带宽绑定实例列表。 数组长度: 0 - 200

表 4-441 Tag

参数	参数类型	描述
key	String	标签键, 最大长度128个unicode字符, 格式为大小写字母, 数字, 中划线“-”, 下划线“_”, 中文。
value	String	标签值, 最大长度255个unicode字符, 格式为大小写字母, 数字, 中划线“-”, 下划线“_”, 点“.”, 中文。

表 4-442 GlobalConnectionBandwidthAssociatedInstance

参数	参数类型	描述
id	String	绑定实例ID。 最小长度: 0 最大长度: 36
type	String	绑定实例类型。 最小长度: 0 最大长度: 36
region_id	String	绑定实例的region信息, global服务默认取值"global"。 最小长度: 0 最大长度: 36
project_id	String	绑定实例的project id信息。 最小长度: 0 最大长度: 36

## 请求示例

查询全域互联带宽列表

```
GET https://{cc_endpoint}/v3/{domain_id}/gcb/gcbbandwidths
```

## 响应示例

状态码： 200

查询全域互联带宽列表成功。

```
{
  "request_id": "61126320a1802d5c6444f9d2d76526c2",
  "globalconnection_bandwidths": [ {
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name": "name",
    "description": "description",
    "domain_id": "521bb3d98bed4c6c9ee8669bd0620f76",
    "bordercross": false,
    "type": "Region",
    "binding_service": "CC",
    "enterprise_project_id": "0c478f9e-73a4-4c45-b9bc-b2c3bfc0d4c0",
    "charge_mode": "bwd",
    "size": 100,
    "sla_level": "Au",
    "local_site_code": "cn-south-guangzhou",
    "remote_site_code": "cn-north-beijing4",
    "frozen": false,
    "spec_code_id": "",
    "tags": [ ],
    "created_at": "2024-01-24T08:26:41.914Z",
    "updated_at": "2024-01-24T08:26:41.914Z",
    "enable_share": false,
    "instances": [ ]
  } ],
  "page_info": {
    "next_marker": "XXX",
    "previous_marker": "XXX",
    "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListGlobalConnectionBandwidthsSolution {

    public static void main(String[] args) {
```

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListGlobalConnectionBandwidthsRequest request = new ListGlobalConnectionBandwidthsRequest();
request.withBindingService();
request.withType();
request.withChargeMode();
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withId();
request.withName();
request.withEnterpriseProjectId();
try {
    ListGlobalConnectionBandwidthsResponse response =
client.listGlobalConnectionBandwidths(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```



```
request = ListGlobalConnectionBandwidthsRequest()
request.binding_service =
request.type =
request.charge_mode =
request.limit = <limit>
request.marker = "<marker>"
request.id =
request.name =
request.enterprise_project_id =
response = client.list_global_connection_bandwidths(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询全域互联带宽列表成功。

## 错误码

请参见[错误码](#)。

## 4.13.2 创建全域互联带宽

### 功能介绍

创建全域互联带宽。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/gcb/gcbandwidths

表 4-443 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

### 请求参数

表 4-444 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-445 请求 Body 参数

参数	是否必选	参数类型	描述
globalconnection_bandwidth	是	CreateGlobalConnectionBandwidth object	创建全域互联带宽实例的详细信息。

表 4-446 CreateGlobalConnectionBandwidth

参数	是否必选	参数类型	描述
name	是	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
bordercross	是	Boolean	功能说明：全域互联带宽是否跨境，判断依据：带宽是否涉及从中国大陆到其他国家。取值范围：True：跨境；False：非跨境
type	是	String	功能说明：描述带宽类型，对应地理区间的城域、区域、大区、跨区四级： <ul style="list-style-type: none"><li>• TrsArea: 跨区带宽</li><li>• Area: 大区带宽</li><li>• SubArea: 区域带宽</li><li>• Region: 城域带宽</li></ul>
enterprise_project_id	否	String	实例所属企业项目ID。 最小长度：1 最大长度：36
tags	否	Array of Tag objects	实例标签。 数组长度：1 - 20
charge_mode	是	String	功能说明：描述计费类型，描述可选计费类型。默认开放按带宽计费，传统95计费租户白名单控制。取值范围：bwd: 按带宽计费 95: 按传统型95计费

参数	是否必选	参数类型	描述
size	是	Integer	功能说明：全域互联带宽实例中的带宽值大小，单位Mbit/s。 取值范围：2-300Mbit/s 最小值： <b>2</b> 最大值： <b>300</b>
sla_level	否	String	功能说明：描述网络等级，从高到低分为铂金、金、银。默认金，其余租户白名单控制。 <ul style="list-style-type: none"><li>● Pt: 铂金</li><li>● Au: 金</li><li>● Ag: 银</li></ul> 缺省值： <b>Au</b> 最小长度： <b>1</b> 最大长度： <b>36</b>
local_area	否	String	功能说明：本端接入点，配合remote_area信息描述带宽实例应用的范围。取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-(中划线)、.(点)，站点编码通过接口获取，带宽类型为Region可不传，其他类型必传 最小长度： <b>1</b> 最大长度： <b>36</b>
remote_area	否	String	功能说明：远端接入点，配合local_area信息描述带宽实例应用的范围。取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-(中划线)、.(点)，站点编码通过接口获取，带宽类型为Region可不传，其他类型必传 最小长度： <b>1</b> 最大长度： <b>36</b>
spec_code_id	否	String	功能说明：线路规格编码UUID。 最小长度： <b>1</b> 最大长度： <b>36</b>

表 4-447 Tag

参数	是否必选	参数类型	描述
key	是	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	否	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

## 响应参数

状态码： 201

表 4-448 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
globalconnection_bandwidth	<b>GlobalConnectionBandwidth</b> object	全域互联带宽实例的详细信息。

表 4-449 GlobalConnectionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度： 0 最大长度： 255
domain_id	String	实例所属账号ID。
bordercross	Boolean	功能说明：全域互联带宽是否跨境，判断依据：带宽是否涉及从中国大陆到其他国家。取值范围： True：跨境； False：非跨境

参数	参数类型	描述
type	String	功能说明：描述带宽类型，对应地理区间的城域、区域、大区、跨区四级： <ul style="list-style-type: none"><li>• TrsArea: 跨区带宽</li><li>• Area: 大区带宽</li><li>• SubArea: 区域带宽</li><li>• Region: 城域带宽</li></ul>
binding_service	String	功能说明：绑定的服务类型。实例类型： <ul style="list-style-type: none"><li>• CC: 云连接</li><li>• GEIP: 全域弹性公网IP</li><li>• GCN: 中心网络</li><li>• GSN: 分支网络</li><li>• ALL: 所有实例类型</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
charge_mode	String	功能说明：描述计费类型，描述可选计费类型。默认开放按带宽计费，传统95计费租户白名单控制。取值范围：bwd: 按带宽计费 95: 按传统型95计费
size	Integer	功能说明：全域互联带宽实例中的带宽值大小，单位Mbit/s。取值范围：2-300Mbit/s 最小值：2 最大值：300
sla_level	String	功能说明：描述网络等级，从高到低分为铂金、金、银。默认金，其余租户白名单控制。 <ul style="list-style-type: none"><li>• Pt: 铂金</li><li>• Au: 金</li><li>• Ag: 银</li></ul> 缺省值：Au 最小长度：1 最大长度：36
local_area	String	功能说明：本端接入点的中英文名。通过HEADER里面的x-language控制，默认英文，zh-cn返回中文。 最小长度：1 最大长度：36

参数	参数类型	描述
remote_area	String	功能说明: 远端接入点的中英文名。通过HEADER里面的x-language控制, 默认英文, zh-cn返回中文。 最小长度: 1 最大长度: 36
local_site_code	String	功能说明: 本端接入点的编码。 最小长度: 1 最大长度: 36
remote_site_code	String	功能说明: 远端接入点的编码。 最小长度: 1 最大长度: 36
admin_state	String	功能说明: 全域互联带宽状态。取值范围: NORMAL-正常 FREEZED-冻结状态
frozen	Boolean	功能说明: 全域互联带宽是否冻结。取值范围: true-冻结 false-非冻结
spec_code_id	String	功能说明: 线路规格编码UUID。 最小长度: 1 最大长度: 36
tags	Array of <b>Tag</b> objects	实例标签。 数组长度: 1 - 20
created_at	String	实例创建时间。UTC时间格式, yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式, yyyy-MM-ddTHH:mm:ss。
enable_share	Boolean	功能说明: 全域互联带宽是否支持绑定多实例。 取值范围: true-支持 false-不支持
instances	Array of <b>GlobalConnectionBandwidthAssociateInstance</b> objects	功能说明: 全域互联带宽绑定实例列表。 数组长度: 0 - 200

表 4-450 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-451 GlobalConnectionBandwidthAssociatedInstance

参数	参数类型	描述
id	String	绑定实例ID。 最小长度：0 最大长度：36
type	String	绑定实例类型。 最小长度：0 最大长度：36
region_id	String	绑定实例的region信息，global服务默认取值"global"。 最小长度：0 最大长度：36
project_id	String	绑定实例的project id信息。 最小长度：0 最大长度：36

## 请求示例

- 创建城域带宽

```
POST https://{cc_endpoint}/v3/{domain_id}/gcb/gcbandwidths
```

```
{
  "globalconnection_bandwidth": {
    "name": "test_region",
    "description": "Description",
    "bordercross": false,
    "type": "Region",
    "charge_mode": "bwd",
    "size": 5
  }
}
```

- 创建大区带宽

```
POST https://{cc_endpoint}/v3/{domain_id}/gcb/gcbandwidth
```

```
{
```



```
"globalconnection_bandwidth" : {
  "name" : "test_area",
  "description" : "Description",
  "bordercross" : false,
  "type" : "Area",
  "charge_mode" : "bwd",
  "size" : 5,
  "local_area" : "cn-north-beijing4",
  "remote_area" : "cn-south-guangzhou"
}
```

## 响应示例

**状态码： 201**

创建全域互联带宽成功。

```
{
  "request_id" : "1b90e4762e3090961a30ca3a712dc0ed",
  "globalconnection_bandwidth" : {
    "id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name" : "name",
    "description" : "description",
    "domain_id" : "521bb3d98bed4c6c9ee8669bd0620f76",
    "bordercross" : false,
    "type" : "Region",
    "binding_service" : "CC",
    "enterprise_project_id" : "0c478f9e-73a4-4c45-b9bc-b2c3bfc0d4c0",
    "charge_mode" : "bwd",
    "size" : 100,
    "sla_level" : "Au",
    "local_site_code" : "cn-south-guangzhou",
    "remote_site_code" : "cn-north-beijing4",
    "frozen" : false,
    "spec_code_id" : "",
    "tags" : [ ],
    "created_at" : "2024-01-24T08:26:41.914Z",
    "updated_at" : "2024-01-24T08:26:41.914Z",
    "enable_share" : false,
    "instances" : [ ]
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 创建城域带宽

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class CreateGlobalConnectionBandwidthSolution {

  public static void main(String[] args) {
```

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
CreateGlobalConnectionBandwidthRequest request = new
CreateGlobalConnectionBandwidthRequest();
CreateGlobalConnectionBandwidthRequestBody body = new
CreateGlobalConnectionBandwidthRequestBody();
CreateGlobalConnectionBandwidth globalconnectionBandwidthbody = new
CreateGlobalConnectionBandwidth();
globalconnectionBandwidthbody.withSize(5)
    .withName("test_region")
    .withDescription("Description")
    .withChargeMode(CreateGlobalConnectionBandwidth.ChargeModeEnum.fromValue("bwd"))
    .withType(CreateGlobalConnectionBandwidth.TypeEnum.fromValue("Region"))
    .withBordercross(false);
body.withGlobalconnectionBandwidth(globalconnectionBandwidthbody);
request.withBody(body);
try {
    CreateGlobalConnectionBandwidthResponse response =
client.createGlobalConnectionBandwidth(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- **创建大区带宽**

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class CreateGlobalConnectionBandwidthSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
CreateGlobalConnectionBandwidthRequest request = new
CreateGlobalConnectionBandwidthRequest();
CreateGlobalConnectionBandwidthRequestBody body = new
CreateGlobalConnectionBandwidthRequestBody();
CreateGlobalConnectionBandwidth globalconnectionBandwidthbody = new
CreateGlobalConnectionBandwidth();
globalconnectionBandwidthbody.setSize(5)
    .withName("test_area")
    .withDescription("Description")
    .withChargeMode(CreateGlobalConnectionBandwidth.ChargeModeEnum.fromValue("bwd"))
    .withType(CreateGlobalConnectionBandwidth.TypeEnum.fromValue("Area"))
    .withBordercross(false)
    .withLocalArea("cn-north-beijing4")
    .withRemoteArea("cn-south-guangzhou");
body.withGlobalconnectionBandwidth(globalconnectionBandwidthbody);
request.withBody(body);
try {
    CreateGlobalConnectionBandwidthResponse response =
client.createGlobalConnectionBandwidth(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

- 创建城域带宽

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

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

```
.with_credentials(credentials) \  
.with_region(CcRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = CreateGlobalConnectionBandwidthRequest()  
    globalconnectionbandwidthbody = CreateGlobalConnectionBandwidth(  
        size=5,  
        name="test_region",  
        description="Description",  
        charge_mode="bwd",  
        type="Region",  
        bordercross=False  
    )  
    request.body = CreateGlobalConnectionBandwidthRequestBody(  
        globalconnection_bandwidth=globalconnectionbandwidthbody  
    )  
    response = client.create_global_connection_bandwidth(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

- **创建大区带宽**

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkcc.v3.region.cc_region import CcRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcc.v3 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    # environment variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CcRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateGlobalConnectionBandwidthRequest()  
        globalconnectionbandwidthbody = CreateGlobalConnectionBandwidth(  
            size=5,  
            name="test_area",  
            description="Description",  
            charge_mode="bwd",  
            type="Area",  
            bordercross=False,  
            local_area="cn-north-beijing4",  
            remote_area="cn-south-guangzhou"  
        )  
        request.body = CreateGlobalConnectionBandwidthRequestBody(  
            globalconnection_bandwidth=globalconnectionbandwidthbody  
        )  
        response = client.create_global_connection_bandwidth(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)
```

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

## Go

- 创建城域带宽

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateGlobalConnectionBandwidthRequest{
        descriptionGlobalconnectionBandwidth:= "Description"
        globalconnectionBandwidthbody := &model.CreateGlobalConnectionBandwidth{
            Size: int32(5),
            Name: "test_region",
            Description: &descriptionGlobalconnectionBandwidth,
            ChargeMode: model.GetCreateGlobalConnectionBandwidthChargeModeEnum().BWD,
            Type: model.GetCreateGlobalConnectionBandwidthTypeEnum().REGION,
            Bordercross: false,
        }
    }
    request.Body = &model.CreateGlobalConnectionBandwidthRequestBody{
        GlobalconnectionBandwidth: globalconnectionBandwidthbody,
    }
    response, err := client.CreateGlobalConnectionBandwidth(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 创建大区带宽

```
package main

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

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateGlobalConnectionBandwidthRequest{
        descriptionGlobalconnectionBandwidth:= "Description"
        localAreaGlobalconnectionBandwidth:= "cn-north-beijing4"
        remoteAreaGlobalconnectionBandwidth:= "cn-south-guangzhou"
        globalconnectionBandwidthbody := &model.CreateGlobalConnectionBandwidth{
            Size: int32(5),
            Name: "test_area",
            Description: &descriptionGlobalconnectionBandwidth,
            ChargeMode: model.GetCreateGlobalConnectionBandwidthChargeModeEnum().BWD,
            Type: model.GetCreateGlobalConnectionBandwidthTypeEnum().AREA,
            Bordercross: false,
            LocalArea: &localAreaGlobalconnectionBandwidth,
            RemoteArea: &remoteAreaGlobalconnectionBandwidth,
        }
    }
    request.Body = &model.CreateGlobalConnectionBandwidthRequestBody{
        GlobalconnectionBandwidth: globalconnectionBandwidthbody,
    }
    response, err := client.CreateGlobalConnectionBandwidth(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
201	创建全域互联带宽成功。

## 错误码

请参见[错误码](#)。

### 4.13.3 查询全域互联带宽详情

#### 功能介绍

查询全域互联带宽详情。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{domain\_id}/gcb/gcbbandwidths/{id}

表 4-452 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

#### 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-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

#### 响应参数

状态码：200

表 4-454 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
globalconnection_bandwidth	<b>GlobalConnectionBandwidth</b> object	全域互联带宽实例的详细信息。

表 4-455 GlobalConnectionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
bordercross	Boolean	功能说明：全域互联带宽是否跨境，判断依据：带宽是否涉及从中国大陆到其他国家。取值范围：True：跨境；False：非跨境
type	String	功能说明：描述带宽类型，对应地理区间的城域、区域、大区、跨区四级： <ul style="list-style-type: none"><li>• TrsArea: 跨区带宽</li><li>• Area: 大区带宽</li><li>• SubArea: 区域带宽</li><li>• Region: 城域带宽</li></ul>
binding_service	String	功能说明：绑定的服务类型。实例类型： <ul style="list-style-type: none"><li>• CC: 云连接</li><li>• GEIP: 全域弹性公网IP</li><li>• GCN: 中心网络</li><li>• GSN: 分支网络</li><li>• ALL: 所有实例类型</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36



参数	参数类型	描述
charge_mode	String	功能说明: 描述计费类型, 描述可选计费类型。默认开放按带宽计费, 传统95计费租户白名单控制。取值范围: bwd: 按带宽计费 95: 按传统型95计费
size	Integer	功能说明: 全域互联带宽实例中的带宽值大小, 单位Mbit/s。取值范围: 2-300Mbit/s 最小值: 2 最大值: 300
sla_level	String	功能说明: 描述网络等级, 从高到低分为铂金、金、银。默认金, 其余租户白名单控制。 <ul style="list-style-type: none"><li>● Pt: 铂金</li><li>● Au: 金</li><li>● Ag: 银</li></ul> 缺省值: <b>Au</b> 最小长度: 1 最大长度: 36
local_area	String	功能说明: 本端接入点的中英文名。通过HEADER里面的x-language控制, 默认英文, zh-cn返回中文。 最小长度: 1 最大长度: 36
remote_area	String	功能说明: 远端接入点的中英文名。通过HEADER里面的x-language控制, 默认英文, zh-cn返回中文。 最小长度: 1 最大长度: 36
local_site_code	String	功能说明: 本端接入点的编码。 最小长度: 1 最大长度: 36
remote_site_code	String	功能说明: 远端接入点的编码。 最小长度: 1 最大长度: 36
admin_state	String	功能说明: 全域互联带宽状态。取值范围: NORMAL-正常 FREEZED-冻结状态
frozen	Boolean	功能说明: 全域互联带宽是否冻结。取值范围: true-冻结 false-非冻结

参数	参数类型	描述
spec_code_id	String	功能说明：线路规格编码UUID。 最小长度：1 最大长度：36
tags	Array of <b>Tag</b> objects	实例标签。 数组长度：1 - 20
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
enable_share	Boolean	功能说明：全域互联带宽是否支持绑定多实例。 取值范围：true-支持 false-不支持
instances	Array of <b>GlobalConnectionBandwidthAssociateInstance</b> objects	功能说明：全域互联带宽绑定实例列表。 数组长度：0 - 200

表 4-456 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-457 GlobalConnectionBandwidthAssociatedInstance

参数	参数类型	描述
id	String	绑定实例ID。 最小长度：0 最大长度：36
type	String	绑定实例类型。 最小长度：0 最大长度：36

参数	参数类型	描述
region_id	String	绑定实例的region信息，global服务默认取值"global"。 最小长度：0 最大长度：36
project_id	String	绑定实例的project id信息。 最小长度：0 最大长度：36

## 请求示例

查询全域互联带宽详情

```
GET https://{cc_endpoint}/v3/{domain_id}/gcb/gcbbandwidths/{id}
```

## 响应示例

状态码：200

查询全域互联带宽详情成功。

```
{
  "request_id": "1b90e4762e3090961a30ca3a712dc0ed",
  "globalconnection_bandwidth": {
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name": "name",
    "description": "description",
    "domain_id": "521bb3d98bed4c6c9ee8669bd0620f76",
    "bordercross": false,
    "type": "Region",
    "binding_service": "CC",
    "enterprise_project_id": "0c478f9e-73a4-4c45-b9bc-b2c3bfc0d4c0",
    "charge_mode": "bwd",
    "size": 100,
    "sla_level": "Au",
    "local_site_code": "cn-south-guangzhou",
    "remote_site_code": "cn-north-beijing4",
    "frozen": false,
    "spec_code_id": "",
    "tags": [],
    "created_at": "2024-01-24T08:26:41.914Z",
    "updated_at": "2024-01-24T08:26:41.914Z",
    "enable_share": false,
    "instances": []
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ShowGlobalConnectionBandwidthSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowGlobalConnectionBandwidthRequest request = new ShowGlobalConnectionBandwidthRequest();
        try {
            ShowGlobalConnectionBandwidthResponse response =
                client.showGlobalConnectionBandwidth(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = ShowGlobalConnectionBandwidthRequest()
    response = client.show_global_connection_bandwidth(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询全域互联带宽详情成功。

## 错误码

请参见[错误码](#)。

## 4.13.4 更新全域互联带宽详情

### 功能介绍

更新全域互联带宽详情。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{domain\_id}/gcb/gcbbandwidths/{id}

表 4-458 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

### 请求参数

表 4-459 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-460 请求 Body 参数

参数	是否必选	参数类型	描述
globalconnection_bandwidth	是	UpdateGlobalConnectionBandwidth object	更新全域互联带宽的详细信息

表 4-461 UpdateGlobalConnectionBandwidth

参数	是否必选	参数类型	描述
name	否	String	实例名字。
description	否	String	实例描述。不支持 <>。 最小长度： <b>0</b> 最大长度： <b>255</b>
size	否	Integer	功能说明：全域互联带宽实例中的带宽值大小，单位Mbit/s。 取值范围：2-300Mbit/s 最小值： <b>2</b> 最大值： <b>300</b>
charge_mode	否	String	功能说明：描述计费类型，描述可选计费类型。默认开放按带宽计费，传统95计费租户白名单控制。取值范围：bwd: 按带宽计费 95: 按传统型95计费
sla_level	否	String	功能说明：描述网络等级，从高到低分为铂金、金、银。默认金，其余租户白名单控制。 <ul style="list-style-type: none"><li>● Pt: 铂金</li><li>● Au: 金</li><li>● Ag: 银</li></ul> 缺省值： <b>Au</b> 最小长度： <b>1</b> 最大长度： <b>36</b>
binding_service	否	String	功能说明：绑定的服务类型。实例类型： <ul style="list-style-type: none"><li>● CC: 云连接</li><li>● GEIP: 全域弹性公网IP</li><li>● GCN: 中心网络</li><li>● GSN: 分支网络</li><li>● ALL: 所有实例类型</li></ul>

参数	是否必选	参数类型	描述
spec_code_id	否	String	功能说明：线路规格编码UUID。 最小长度：1 最大长度：36

## 响应参数

状态码：200

表 4-462 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
globalconnection_bandwidth	<b>GlobalConnectionBandwidth</b> object	全域互联带宽实例的详细信息。

表 4-463 GlobalConnectionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
bordercross	Boolean	功能说明：全域互联带宽是否跨境，判断依据：带宽是否涉及从中国大陆到其他国家。取值范围：True：跨境；False：非跨境
type	String	功能说明：描述带宽类型，对应地理区间的城域、区域、大区、跨区四级： <ul style="list-style-type: none"><li>• TrsArea: 跨区带宽</li><li>• Area: 大区带宽</li><li>• SubArea: 区域带宽</li><li>• Region: 城域带宽</li></ul>



参数	参数类型	描述
binding_service	String	功能说明：绑定的服务类型。实例类型： <ul style="list-style-type: none"><li>● CC: 云连接</li><li>● GEIP: 全域弹性公网IP</li><li>● GCN: 中心网络</li><li>● GSN: 分支网络</li><li>● ALL: 所有实例类型</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
charge_mode	String	功能说明：描述计费类型，描述可选计费类型。默认开放按带宽计费，传统95计费租户白名单控制。取值范围：bwd: 按带宽计费 95: 按传统型95计费
size	Integer	功能说明：全域互联带宽实例中的带宽值大小，单位Mbit/s。取值范围：2-300Mbit/s 最小值：2 最大值：300
sla_level	String	功能说明：描述网络等级，从高到低分为铂金、金、银。默认金，其余租户白名单控制。 <ul style="list-style-type: none"><li>● Pt: 铂金</li><li>● Au: 金</li><li>● Ag: 银</li></ul> 缺省值：Au 最小长度：1 最大长度：36
local_area	String	功能说明：本端接入点的中英文名。通过HEADER里面的x-language控制，默认英文，zh-cn返回中文。 最小长度：1 最大长度：36
remote_area	String	功能说明：远端接入点的中英文名。通过HEADER里面的x-language控制，默认英文，zh-cn返回中文。 最小长度：1 最大长度：36
local_site_code	String	功能说明：本端接入点的编码。 最小长度：1 最大长度：36

参数	参数类型	描述
remote_site_code	String	功能说明: 远端接入点的编码。 最小长度: <b>1</b> 最大长度: <b>36</b>
admin_state	String	功能说明: 全域互联带宽状态。取值范围: NORMAL-正常 FREEZED-冻结状态
frozen	Boolean	功能说明: 全域互联带宽是否冻结。取值范围: true-冻结 false-非冻结
spec_code_id	String	功能说明: 线路规格编码UUID。 最小长度: <b>1</b> 最大长度: <b>36</b>
tags	Array of <b>Tag</b> objects	实例标签。 数组长度: <b>1 - 20</b>
created_at	String	实例创建时间。UTC时间格式, yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式, yyyy-MM-ddTHH:mm:ss。
enable_share	Boolean	功能说明: 全域互联带宽是否支持绑定多实例。 取值范围: true-支持 false-不支持
instances	Array of <b>GlobalConnectionBandwidthAssociateInstance</b> objects	功能说明: 全域互联带宽绑定实例列表。 数组长度: <b>0 - 200</b>

表 4-464 Tag

参数	参数类型	描述
key	String	标签键, 最大长度128个unicode字符, 格式为大小写字母, 数字, 中划线“-”, 下划线“_”, 中文。
value	String	标签值, 最大长度255个unicode字符, 格式为大小写字母, 数字, 中划线“-”, 下划线“_”, 点“.”, 中文。

表 4-465 GlobalConnectionBandwidthAssociatedInstance

参数	参数类型	描述
id	String	绑定实例ID。 最小长度：0 最大长度：36
type	String	绑定实例类型。 最小长度：0 最大长度：36
region_id	String	绑定实例的region信息，global服务默认取值"global"。 最小长度：0 最大长度：36
project_id	String	绑定实例的project id信息。 最小长度：0 最大长度：36

## 请求示例

更新全域互联带宽

```
PUT https://{cc_endpoint}/v3/{domain_id}/gcb/gcbandwidths/{id}
```

```
{
  "globalconnection_bandwidth" : {
    "name" : "test",
    "description" : "Description",
    "size" : 5
  }
}
```

## 响应示例

状态码：200

更新全域互联带宽详情成功。

```
{
  "request_id" : "1b90e4762e3090961a30ca3a712dc0ed",
  "globalconnection_bandwidth" : {
    "id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
    "name" : "name",
    "description" : "description",
    "domain_id" : "521bb3d98bed4c6c9ee8669bd0620f76",
    "bordercross" : false,
    "type" : "Region",
    "binding_service" : "CC",
    "enterprise_project_id" : "0c478f9e-73a4-4c45-b9bc-b2c3bfc0d4c0",
    "charge_mode" : "bwd",
    "size" : 100,
    "sla_level" : "Au",
    "local_site_code" : "cn-south-guangzhou",
    "remote_site_code" : "cn-north-beijing4",
  }
}
```

```
"frozen" : "false",  
"spec_code_id" : "",  
"tags" : [],  
"created_at" : "2024-01-24T08:26:41.914Z",  
"updated_at" : "2024-01-24T08:26:41.914Z",  
"enable_share" : false,  
"instances" : []  
}  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

#### 更新全域互联带宽

```
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.cc.v3.region.CcRegion;  
import com.huaweicloud.sdk.cc.v3.*;  
import com.huaweicloud.sdk.cc.v3.model.*;  
  
public class UpdateGlobalConnectionBandwidthSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CcClient client = CcClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateGlobalConnectionBandwidthRequest request = new  
UpdateGlobalConnectionBandwidthRequest();  
        UpdateGlobalConnectionBandwidthRequestBody body = new  
UpdateGlobalConnectionBandwidthRequestBody();  
        UpdateGlobalConnectionBandwidth globalconnectionBandwidthbody = new  
UpdateGlobalConnectionBandwidth();  
        globalconnectionBandwidthbody.setSize(5)  
            .WithName("test")  
            .withDescription("Description");  
        body.withGlobalconnectionBandwidth(globalconnectionBandwidthbody);  
        request.withBody(body);  
        try {  
            UpdateGlobalConnectionBandwidthResponse response =  
client.updateGlobalConnectionBandwidth(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {
```

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

## Python

### 更新全域互联带宽

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateGlobalConnectionBandwidthRequest()
        globalconnectionbandwidthbody = UpdateGlobalConnectionBandwidth(
            size=5,
            name="test",
            description="Description"
        )
        request.body = UpdateGlobalConnectionBandwidthRequestBody(
            globalconnection_bandwidth=globalconnectionbandwidthbody
        )
        response = client.update_global_connection_bandwidth(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 更新全域互联带宽

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
```

```
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment 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 := cc.NewCcClient(  
        cc.CcClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.UpdateGlobalConnectionBandwidthRequest{  
        sizeGlobalconnectionBandwidth:= int32(5)  
        nameGlobalconnectionBandwidth:= "test"  
        descriptionGlobalconnectionBandwidth:= "Description"  
        globalconnectionBandwidthbody := &model.UpdateGlobalConnectionBandwidth{  
            Size: &sizeGlobalconnectionBandwidth,  
            Name: &nameGlobalconnectionBandwidth,  
            Description: &descriptionGlobalconnectionBandwidth,  
        }  
    }  
    request.Body = &model.UpdateGlobalConnectionBandwidthRequestBody{  
        GlobalconnectionBandwidth: globalconnectionBandwidthbody,  
    }  
    response, err := client.UpdateGlobalConnectionBandwidth(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	更新全域互联带宽详情成功。

## 错误码

请参见[错误码](#)。

## 4.13.5 删除全域互联带宽

### 功能介绍

删除全域互联带宽。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/gcb/gcbandwidths/{id}

表 4-466 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

### 请求参数

表 4-467 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

### 响应参数

无

### 请求示例

删除全域互联带宽

```
DELETE https://{cc_endpoint}/v3/{domain_id}/gcb/gcbandwidths/{id}
```

## 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class DeleteGlobalConnectionBandwidthSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteGlobalConnectionBandwidthRequest request = new
        DeleteGlobalConnectionBandwidthRequest();
        try {
            DeleteGlobalConnectionBandwidthResponse response =
            client.deleteGlobalConnectionBandwidth(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
```



```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteGlobalConnectionBandwidthRequest()
        response = client.delete_global_connection_bandwidth(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
204	删除全域互联带宽成功。

## 错误码

请参见[错误码](#)。

## 4.13.6 全域互联带宽绑定实例

### 功能介绍

全域互联带宽绑定实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/gcb/gcbandwidths/{id}/associate-instance

表 4-468 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36

## 请求参数

表 4-469 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-470 请求 Body 参数

参数	是否必选	参数类型	描述
gcbandwidths	是	Array of <a href="#">AssociateGlobalConnectionBandwidthInstanceRequestInfo</a> objects	全域互联带宽绑定实例详细信息。 数组长度：1 - 200

表 4-471 AssociateGlobalConnectionBandwidthInstanceRequestInfo

参数	是否必选	参数类型	描述
resource_id	是	String	功能说明：实例ID。取值范围：1-36个字符，支持数字、字母、_(下划线)、-(中划线) 最小长度：1 最大长度：36
resource_type	是	String	功能说明：实例类型。 最小长度：1 最大长度：36
region_id	否	String	功能说明：实例所在region，不填默认"global"。 最小长度：1 最大长度：36
project_id	是	String	功能说明：实例所在region对应的projectId。 最小长度：1 最大长度：36

## 响应参数

状态码： 200

表 4-472 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
gcbandwidths	Array of AssociateGlobalConnectionBandwidthInstanceResponseInfo objects	全域互联带宽绑定实例响应详情。 数组长度： 1 - 200

表 4-473 AssociateGlobalConnectionBandwidthInstanceResponseInfo

参数	参数类型	描述
resource_id	String	功能说明：实例ID。取值范围：1-36个字符，支持数字、字母、_(下划线)、-(中划线) 最小长度： 1 最大长度： 36
resource_type	String	功能说明：实例类型。 最小长度： 1 最大长度： 36
region_id	String	功能说明：实例所在region，不填默认"global"。 最小长度： 1 最大长度： 36
project_id	String	功能说明：实例所在region对应的projectId。 最小长度： 1 最大长度： 36
result	String	功能说明：绑定操作成功还是失败。 <ul style="list-style-type: none"><li>• success: 成功</li><li>• fail: 失败</li></ul>
message	String	功能说明：绑定操作如果失败，具体的错误信息。 最小长度： 0 最大长度： 4096

## 请求示例

### 全域互联带宽绑定实例

```
POST https://{cc_endpoint}/v3/{domain_id}/gcb/gcbbandwidths/{id}/associate-instance
{
  "gcbbandwidths": [ {
    "resource_type": "GEIP",
    "resource_id": "37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc",
    "project_id": "XXX",
    "region_id": "global"
  } ]
}
```

## 响应示例

### 状态码： 200

全域互联带宽绑定实例成功。

```
{
  "request_id": "1b90e4762e3090961a30ca3a712dc0ed",
  "gcbbandwidths": [ {
    "resource_id": "37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc",
    "resource_type": "GEIP",
    "project_id": "XXX",
    "region_id": "global",
    "result": "success",
    "message": ""
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

#### 全域互联带宽绑定实例

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class AssociateGlobalConnectionBandwidthInstanceSolution {

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

```
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
AssociateGlobalConnectionBandwidthInstanceRequest request = new
AssociateGlobalConnectionBandwidthInstanceRequest();
AssociateGlobalConnectionBandwidthInstanceRequestBody body = new
AssociateGlobalConnectionBandwidthInstanceRequestBody();
List<AssociateGlobalConnectionBandwidthInstanceRequestInfo> listbodyGcbandwidths = new
ArrayList<>();
listbodyGcbandwidths.add(
    new AssociateGlobalConnectionBandwidthInstanceRequestInfo()
        .withResourceType("GEIP")
        .withRegionId("global")
        .withResourceId("37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc")
        .withProjectId("XXX")
    );
body.withGcbandwidths(listbodyGcbandwidths);
request.withBody(body);
try {
    AssociateGlobalConnectionBandwidthInstanceResponse response =
client.associateGlobalConnectionBandwidthInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 全域互联带宽绑定实例

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

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

```
.with_region(CcRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = AssociateGlobalConnectionBandwidthInstanceRequest()
    listGcbandwidthsbody = [
        AssociateGlobalConnectionBandwidthInstanceRequestInfo(
            resource_type="GEIP",
            region_id="global",
            resource_id="37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc",
            project_id="XXX"
        )
    ]
    request.body = AssociateGlobalConnectionBandwidthInstanceRequestBody(
        gcbandwidths=listGcbandwidthsbody
    )
    response = client.associate_global_connection_bandwidth_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 全域互联带宽绑定实例

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AssociateGlobalConnectionBandwidthInstanceRequest{}
    regionIdGcbandwidths:= "global"
    var listGcbandwidthsbody = []model.AssociateGlobalConnectionBandwidthInstanceRequestInfo{
        {
            ResourceType: "GEIP",
            RegionId: &regionIdGcbandwidths,
            ResourceId: "37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc",
            ProjectId: "XXX",
        },
    }
    request.Body = &model.AssociateGlobalConnectionBandwidthInstanceRequestBody{
```

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

## 更多

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

## 状态码

状态码	描述
200	全域互联带宽绑定实例成功。

## 错误码

请参见[错误码](#)。

## 4.13.7 全域互联带宽解绑实例

### 功能介绍

全域互联带宽解绑实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/gcb/gcbandwidths/{id}/disassociate-instance

表 4-474 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32
id	是	String	资源的Id。 最小长度：32 最大长度：36



## 请求参数

表 4-475 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

表 4-476 请求 Body 参数

参数	是否必选	参数类型	描述
gcbandwidths	是	Array of <a href="#">DisassociateGlobalConnectionBandwidthInstanceRequestInfo</a> objects	全域互联带宽解绑实例详细信息。 数组长度：1 - 200

表 4-477 DisassociateGlobalConnectionBandwidthInstanceRequestInfo

参数	是否必选	参数类型	描述
resource_id	是	String	功能说明：实例ID。取值范围：1-36个字符，支持数字、字母、_(下划线)、-(中划线) 最小长度：1 最大长度：36
resource_type	是	String	功能说明：实例类型。 最小长度：1 最大长度：36
region_id	否	String	功能说明：实例所在region，不填默认"global"。 最小长度：1 最大长度：36
project_id	是	String	功能说明：实例所在region对应的projectId。 最小长度：1 最大长度：36

## 响应参数

状态码： 200

表 4-478 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度： 32 最大长度： 36
gcbandwidths	Array of <a href="#">DisassociateGlobalConnectionBandwidthInstanceResponseInfo</a> objects	全域互联带宽解绑实例响应详情。 数组长度： 1 - 200

表 4-479 DisassociateGlobalConnectionBandwidthInstanceResponseInfo

参数	参数类型	描述
resource_id	String	功能说明：实例ID。取值范围：1-36个字符，支持数字、字母、_(下划线)、-(中划线) 最小长度： 1 最大长度： 36
resource_type	String	功能说明：实例类型。 最小长度： 1 最大长度： 36
region_id	String	功能说明：实例所在region，不填默认"global"。 最小长度： 1 最大长度： 36
project_id	String	功能说明：实例所在region对应的projectId。 最小长度： 1 最大长度： 36
result	String	功能说明：解绑操作成功还是失败。 <ul style="list-style-type: none"><li>• success: 成功</li><li>• fail: 失败</li></ul>
message	String	功能说明：解绑操作如果失败，具体的错误信息。 最小长度： 0 最大长度： 4096

## 请求示例

### 全域互联带宽解绑实例

```
POST https://{cc_endpoint}/v3/{domain_id}/gcb/gcbbandwidths/{id}/disassociate-instance

{
  "gcbbandwidths": [ {
    "resource_type": "GEIP",
    "resource_id": "37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc",
    "project_id": "XXX",
    "region_id": "global"
  } ]
}
```

## 响应示例

### 状态码： 200

全域互联带宽解绑实例成功。

```
{
  "request_id": "1b90e4762e3090961a30ca3a712dc0ed",
  "gcbbandwidths": [ {
    "resource_id": "37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc",
    "resource_type": "GEIP",
    "project_id": "XXX",
    "region_id": "global",
    "result": "success",
    "message": ""
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

#### 全域互联带宽解绑实例

```
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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class DisassociateGlobalConnectionBandwidthInstanceSolution {

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

```
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
DisassociateGlobalConnectionBandwidthInstanceRequest request = new
DisassociateGlobalConnectionBandwidthInstanceRequest();
DisassociateGlobalConnectionBandwidthInstanceRequestBody body = new
DisassociateGlobalConnectionBandwidthInstanceRequestBody();
List<DisassociateGlobalConnectionBandwidthInstanceRequestInfo> listbodyGcbandwidths = new
ArrayList<>();
listbodyGcbandwidths.add(
    new DisassociateGlobalConnectionBandwidthInstanceRequestInfo()
        .withResourceType("GEIP")
        .withRegionId("global")
        .withResourceId("37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc")
        .withProjectId("XXX")
    );
body.withGcbandwidths(listbodyGcbandwidths);
request.withBody(body);
try {
    DisassociateGlobalConnectionBandwidthInstanceResponse response =
client.disassociateGlobalConnectionBandwidthInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 全域互联带宽解绑实例

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

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

```
.with_region(CcRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = DisassociateGlobalConnectionBandwidthInstanceRequest()
    listGcbandwidthsbody = [
        DisassociateGlobalConnectionBandwidthInstanceRequestInfo(
            resource_type="GEIP",
            region_id="global",
            resource_id="37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc",
            project_id="XXX"
        )
    ]
    request.body = DisassociateGlobalConnectionBandwidthInstanceRequestBody(
        gcbandwidths=listGcbandwidthsbody
    )
    response = client.disassociate_global_connection_bandwidth_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

### 全域互联带宽解绑实例

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisassociateGlobalConnectionBandwidthInstanceRequest{}
    regionIdGcbandwidths := "global"
    var listGcbandwidthsbody = []model.DisassociateGlobalConnectionBandwidthInstanceRequestInfo{
        {
            ResourceType: "GEIP",
            RegionId: &regionIdGcbandwidths,
            ResourceId: "37289f8a-8fd2-4eff-a6c4-f3b94bdbb1bc",
            ProjectId: "XXX",
        },
    }
    request.Body = &model.DisassociateGlobalConnectionBandwidthInstanceRequestBody{
```

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

## 更多

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

## 状态码

状态码	描述
200	全域互联带宽解绑实例成功。

## 错误码

请参见[错误码](#)。

## 4.13.8 查询符合绑定条件的全域互联带宽列表

### 功能介绍

查询符合绑定条件的全域互联带宽列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcb/gcbandwidths/support-bindings

表 4-480 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-481 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值： <b>1</b> 最大值： <b>1000</b> 缺省值： <b>1000</b>
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度： <b>1</b> 最大长度： <b>4096</b>
enterprise_project_id	否	Array	根据企业项目ID过滤列表。 数组长度： <b>1 - 20</b>
local_area	否	String	功能说明：本端接入点。如果是region类型，则返回所有满足条件的城域网带宽，不进行该字段的匹配过滤 如果是其他类型，则会用该字段跟全域互联带宽的local_area进行匹配过滤 附带过滤条件：会通过local_area和remote_area推算最佳全域互联带宽类型进行过滤查询 限制：local_area和remote_area同为空或者同不为空，且站点类型需一致 最小长度： <b>1</b> 最大长度： <b>64</b>

参数	是否必选	参数类型	描述
remote_area	否	String	功能说明：远端接入点。如果是region类型，则返回所有满足条件的城域网带宽，不进行该字段的匹配过滤 如果是其他类型，则会用该字段跟全域互联带宽的remote_area进行匹配过滤 附带过滤条件：会通过local_area和remote_area推算最佳全域互联带宽类型进行过滤查询 限制：local_area和remote_area同为空或者同不为空，且站点类型需一致 最小长度：1 最大长度：64
binding_service	否	String	根据支持绑定实例类型过滤全域互联带宽列表。实例类型： <ul style="list-style-type: none"><li>● CC: 云连接</li><li>● GEIP: 全域弹性公网IP</li><li>● GCN: 中心网络</li><li>● GSN: 分支网络</li></ul>

## 请求参数

表 4-482 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-483 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36



参数	参数类型	描述
page_info	<b>PageInfo</b> object	分页查询页的信息。
globalconnection_bandwidths	Array of <b>GlobalConnectionBandwidth</b> objects	全域互联带宽列表响应体。 数组长度：1 - 2000

表 4-484 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-485 GlobalConnectionBandwidth

参数	参数类型	描述
id	String	资源ID标识符。
name	String	实例名字。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
domain_id	String	实例所属账号ID。
bordercross	Boolean	功能说明：全域互联带宽是否跨境，判断依据：带宽是否涉及从中国大陆到其他国家。取值范围：True：跨境；False：非跨境

参数	参数类型	描述
type	String	功能说明：描述带宽类型，对应地理区间的城域、区域、大区、跨区四级： <ul style="list-style-type: none"><li>• TrsArea: 跨区带宽</li><li>• Area: 大区带宽</li><li>• SubArea: 区域带宽</li><li>• Region: 城域带宽</li></ul>
binding_service	String	功能说明：绑定的服务类型。实例类型： <ul style="list-style-type: none"><li>• CC: 云连接</li><li>• GEIP: 全域弹性公网IP</li><li>• GCN: 中心网络</li><li>• GSN: 分支网络</li><li>• ALL: 所有实例类型</li></ul>
enterprise_project_id	String	实例所属企业项目ID。 最小长度：1 最大长度：36
charge_mode	String	功能说明：描述计费类型，描述可选计费类型。默认开放按带宽计费，传统95计费租户白名单控制。取值范围：bwd: 按带宽计费 95: 按传统型95计费
size	Integer	功能说明：全域互联带宽实例中的带宽值大小，单位Mbit/s。取值范围：2-300Mbit/s 最小值：2 最大值：300
sla_level	String	功能说明：描述网络等级，从高到低分为铂金、金、银。默认金，其余租户白名单控制。 <ul style="list-style-type: none"><li>• Pt: 铂金</li><li>• Au: 金</li><li>• Ag: 银</li></ul> 缺省值：Au 最小长度：1 最大长度：36
local_area	String	功能说明：本端接入点的中英文名。通过HEADER里面的x-language控制，默认英文，zh-cn返回中文。 最小长度：1 最大长度：36

参数	参数类型	描述
remote_area	String	功能说明: 远端接入点的中英文名。通过HEADER里面的x-language控制, 默认英文, zh-cn返回中文。 最小长度: 1 最大长度: 36
local_site_code	String	功能说明: 本端接入点的编码。 最小长度: 1 最大长度: 36
remote_site_code	String	功能说明: 远端接入点的编码。 最小长度: 1 最大长度: 36
admin_state	String	功能说明: 全域互联带宽状态。取值范围: NORMAL-正常 FREEZED-冻结状态
frozen	Boolean	功能说明: 全域互联带宽是否冻结。取值范围: true-冻结 false-非冻结
spec_code_id	String	功能说明: 线路规格编码UUID。 最小长度: 1 最大长度: 36
tags	Array of <b>Tag</b> objects	实例标签。 数组长度: 1 - 20
created_at	String	实例创建时间。UTC时间格式, yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式, yyyy-MM-ddTHH:mm:ss。
enable_share	Boolean	功能说明: 全域互联带宽是否支持绑定多实例。取值范围: true-支持 false-不支持
instances	Array of <b>GlobalConnectionBandwidthAssociateInstance</b> objects	功能说明: 全域互联带宽绑定实例列表。 数组长度: 0 - 200

表 4-486 Tag

参数	参数类型	描述
key	String	标签键，最大长度128个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，中文。
value	String	标签值，最大长度255个unicode字符，格式为大小写字母，数字，中划线“-”，下划线“_”，点“.”，中文。

表 4-487 GlobalConnectionBandwidthAssociatedInstance

参数	参数类型	描述
id	String	绑定实例ID。 最小长度：0 最大长度：36
type	String	绑定实例类型。 最小长度：0 最大长度：36
region_id	String	绑定实例的region信息，global服务默认取值“global”。 最小长度：0 最大长度：36
project_id	String	绑定实例的project id信息。 最小长度：0 最大长度：36

## 请求示例

查询符合绑定条件的全域互联带宽列表

```
GET https://{cc_endpoint}/v3/{domain_id}/gcb/gcbbandwidths/support-bindings?  
binding_service=CC&local_area=cn-south-guangzhou&remote_area=cn-north-beijing4
```

## 响应示例

状态码：200

查询全域互联带宽列表成功。

```
{  
  "request_id": "61126320a1802d5c6444f9d2d76526c2",  
  "globalconnection_bandwidths": [{  
    "id": "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",  
    "name": "name",  
    "description": "description",
```

```
"domain_id" : "521bb3d98bed4c6c9ee8669bd0620f76",
"bordercross" : false,
"type" : "Region",
"binding_service" : "CC",
"enterprise_project_id" : "0c478f9e-73a4-4c45-b9bc-b2c3bfc0d4c0",
"charge_mode" : "bwd",
"size" : 100,
"sla_level" : "Au",
"local_site_code" : "cn-south-guangzhou",
"remote_site_code" : "cn-north-beijing4",
"frozen" : false,
"spec_code_id" : "",
"tags" : [],
"created_at" : "2024-01-24T08:26:41.914Z",
"updated_at" : "2024-01-24T08:26:41.914Z",
"enable_share" : false,
"instances" : []
}],
"page_info" : {
  "next_marker" : "XXX",
  "previous_marker" : "XXX",
  "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListSupportBindingConnectionBandwidthsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSupportBindingConnectionBandwidthsRequest request = new
        ListSupportBindingConnectionBandwidthsRequest();
        request.withLocalArea("<local_area>");
        request.withRemoteArea("<remote_area>");
    }
}
```

```
request.withBindingService(ListSupportBindingConnectionBandwidthsRequest.BindingServiceEnum.fromValue("<binding_service>"));
    request.withLimit(<limit>);
    request.withMarker("<marker>");
    request.withEnterpriseProjectId();
    try {
        ListSupportBindingConnectionBandwidthsResponse response =
client.listSupportBindingConnectionBandwidths(request);
        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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSupportBindingConnectionBandwidthsRequest()
        request.local_area = "<local_area>"
        request.remote_area = "<remote_area>"
        request.binding_service = "<binding_service>"
        request.limit = <limit>
        request.marker = "<marker>"
        request.enterprise_project_id =
        response = client.list_support_binding_connection_bandwidths(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"  
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment 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 := cc.NewCcClient(  
        cc.CcClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListSupportBindingConnectionBandwidthsRequest{}  
    localAreaRequest := "<local_area>"  
    request.LocalArea = &localAreaRequest  
    remoteAreaRequest := "<remote_area>"  
    request.RemoteArea = &remoteAreaRequest  
    request.BindingService =  
model.GetListSupportBindingConnectionBandwidthsRequestBindingServiceEnum().<BINDING_SERVICE>  
    limitRequest := int32(<limit>)  
    request.Limit = &limitRequest  
    markerRequest := "<marker>"  
    request.Marker = &markerRequest  
    response, err := client.ListSupportBindingConnectionBandwidths(request)  
    if err == nil {  
        fmt.Printf("%v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

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

## 状态码

状态码	描述
200	查询全域互联带宽列表成功。

## 错误码

请参见[错误码](#)。

## 4.13.9 查询全域互联带宽租户配置信息

### 功能介绍

查询全域互联带宽租户配置信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcb/configs

表 4-488 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

### 请求参数

表 4-489 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

### 响应参数

状态码：200

表 4-490 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36



参数	参数类型	描述
configs	ListGlobalConnectionBandwidthConfigs object	租户购买全域互联带宽的动态配置项。

表 4-491 ListGlobalConnectionBandwidthConfigs

参数	参数类型	描述
size_range	Array of GlobalConnectionBandwidthSizeRange objects	计费类型对应全域互联带宽大小范围。 数组长度：0 - 10
charge_mode	Array of strings	支持的计费类型列表。 数组长度：0 - 10
services	Array of strings	支持服务实例类型。 数组长度：0 - 10
gcb_type	Array of strings	支持的带宽类型。 数组长度：0 - 10
ratio_95peak_plus	Integer	按增强型95计费保底消费百分比。 最小值：0 最大值：100
ratio_95peak_guar	Integer	按传统型95计费保底消费百分比。 最小值：0 最大值：100
crossborder	Boolean	是否已经完成跨境审批。
quotas	Array of GlobalConnectionBandwidthQuotas objects	配额信息。 数组长度：1 - 10
sla_level	Array of strings	支持线路分级。 数组长度：1 - 10
bind_limit	Integer	共享带宽允许绑定实例数量上限。 最小值：0 最大值：2000
enable_area_bandwidth	Boolean	是否启用传统的大区带宽。

参数	参数类型	描述
enable_change_95	Boolean	是否支持95转按需。
enable_spec_code	Boolean	是否支持多SKU产品功能。

表 4-492 GlobalConnectionBandwidthSizeRange

参数	参数类型	描述
type	String	描述计费类型，描述可选计费类型。取值范围： bwd: 按带宽计费 95: 按传统型95计费 最小长度：0 最大长度：64
min	Integer	全域互联带宽最小值，单位Mbit/s。 最小值：1 最大值：2147483647
max	Integer	全域互联带宽最大值，单位Mbit/s。 最小值：1 最大值：2147483647

表 4-493 GlobalConnectionBandwidthQuotas

参数	参数类型	描述
quota	Integer	配额大小。 最小值：0 最大值：2147483647
used	Integer	已使用值。 最小值：0 最大值：2147483647
type	String	配额类型。gcb.size: 全域互联带宽大小 gcb.count: 全域互联带宽数量

## 请求示例

查询全域互联带宽租户配置信息

```
GET https://{cc_endpoint}/v3/{domain_id}/gcb/configs
```

## 响应示例

**状态码： 200**

查询全域互联带宽详情成功。

```
{
  "request_id": "61126320a1802d5c6444f9d2d76526c2",
  "configs": {
    "size_range": [ {
      "min": 2,
      "max": 300,
      "type": "bwd"
    } ],
    "charge_mode": [ "bwd", 95 ],
    "services": [ "GEIP", "CC", "GCN", "GSN" ],
    "gcb_type": [ "Region", "SubArea", "Area", "TrsArea" ],
    "ratio_95peak_guar": 0,
    "crossborder": false,
    "quotas": [ {
      "quota": 99999999,
      "used": 3451,
      "type": "gcb.size"
    }, {
      "quota": 10000,
      "used": 16,
      "type": "gcb.count"
    } ],
    "bind_limit": 200,
    "sla_level": [ "Pt", "Au", "Ag" ]
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### 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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

public class ListGlobalConnectionBandwidthConfigsSolution {

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

        CcClient client = CcClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(CcRegion.valueOf("<YOUR REGION>"))
        .build();
    ListGlobalConnectionBandwidthConfigsRequest request = new
ListGlobalConnectionBandwidthConfigsRequest();
    try {
        ListGlobalConnectionBandwidthConfigsResponse response =
client.listGlobalConnectionBandwidthConfigs(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListGlobalConnectionBandwidthConfigsRequest()
        response = client.list_global_connection_bandwidth_configs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询全域互联带宽详情成功。

## 错误码

请参见[错误码](#)。

### 4.13.10 查询线路规格列表

#### 功能介绍

查询线路规格列表。租户白名单控制，默认为空。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/gcb/spec-codes

表 4-494 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-495 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
local_area	否	String	线路规格本端接入点编码信息。 最小长度：1 最大长度：64
remote_area	否	String	线路规格远端接入点编码信息。 最小长度：1 最大长度：64

参数	是否必选	参数类型	描述
level	否	Array	带宽等级： <ul style="list-style-type: none"><li>• Pt: 铂金</li><li>• Au: 金</li><li>• Ag: 银</li></ul> 数组长度：1 - 10

## 请求参数

表 4-496 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-497 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
spec_codes	Array of GlobalConnectionBandwidthSpecCode objects	线路规格列表响应体。 数组长度：1 - 10

表 4-498 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-499 GlobalConnectionBandwidthSpecCode

参数	参数类型	描述
id	String	资源ID标识符。
local_area	String	功能说明：本端接入点，配合remote_area信息描述带宽实例应用的范围。取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-(中划线)、.(点)，站点编码通过接口获取，带宽类型为Region可不传，其他类型必传 最小长度：1 最大长度：36
remote_area	String	功能说明：远端接入点，配合local_area信息描述带宽实例应用的范围。取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-(中划线)、.(点)，站点编码通过接口获取，带宽类型为Region可不传，其他类型必传 最小长度：1 最大长度：36
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
name_zh	String	功能说明：线路规格中文名。 最小长度：0 最大长度：64



参数	参数类型	描述
name_en	String	功能说明：线路规格英文名。 最小长度：0 最大长度：64
level	String	支持的线路等级： <ul style="list-style-type: none"><li>● Pt: 铂金</li><li>● Au: 金</li><li>● Ag: 银</li></ul> 最小长度：0 最大长度：64
sku	String	功能描述：GCB特定线路规格产品编码。 最小长度：0 最大长度：64
size	Integer	功能描述：带宽起售值，单位Mbps。 最小值：2 最大值：999999

## 请求示例

查询线路规格列表

```
GET https://{cc_endpoint}/v3/{domain_id}/gcb/spec-codes
```

## 响应示例

状态码：200

查询线路规格列表成功。

```
{
  "request_id": "61126320a1802d5c6444f9d2d76526c2",
  "spec_codes": [ {
    "id": "5c2d5343-cde2-4a41-83ae-56b7f44d73b5",
    "name_zh": "DirectionalBandwidth",
    "name_en": "DirectionalBandwidth",
    "local_area": "cn-southwest-guiyang1",
    "remote_area": "cn-north-wulanchabu1",
    "level": "Ag",
    "sku": "guiyang1_wulanchabu1_ext_ag",
    "size": 2,
    "created_at": "2024-01-24T08:26:41.914Z",
    "updated_at": "2024-01-24T08:26:41.914Z"
  } ],
  "page_info": {
    "next_marker": "XXX",
    "previous_marker": "XXX",
    "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListGlobalConnectionBandwidthSpecCodesSolution {

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

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

        ListGlobalConnectionBandwidthSpecCodesRequest request = new
        ListGlobalConnectionBandwidthSpecCodesRequest();
        request.withLocalArea("<local_area>");
        request.withRemoteArea("<remote_area>");
        request.withLevel();
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        request.withId();
        try {
            ListGlobalConnectionBandwidthSpecCodesResponse response =
            client.listGlobalConnectionBandwidthSpecCodes(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException 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 huaweicloudsdccc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListGlobalConnectionBandwidthSpecCodesRequest()
        request.local_area = "<local_area>"
        request.remote_area = "<remote_area>"
        request.level =
        request.limit = <limit>
        request.marker = "<marker>"
        request.id =
        response = client.list_global_connection_bandwidth_spec_codes(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.ListGlobalConnectionBandwidthSpecCodesRequest{}
localAreaRequest:= "<local_area>"
request.LocalArea = &localAreaRequest
remoteAreaRequest:= "<remote_area>"
request.RemoteArea = &remoteAreaRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
response, err := client.ListGlobalConnectionBandwidthSpecCodes(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	查询线路规格列表成功。

## 错误码

请参见[错误码](#)。

## 4.13.11 查询站点列表

### 功能介绍

查询站点列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/gcb/sites

表 4-500 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-501 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围： 1~1000。 最小值： <b>1</b> 最大值： <b>1000</b> 缺省值： <b>1000</b>
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度： <b>1</b> 最大长度： <b>4096</b>
id	否	Array	根据id查询，可查询多个id。 数组长度： <b>1 - 20</b>
name_en	否	String	站点信息自定义英文名称。 最小长度： <b>1</b> 最大长度： <b>64</b>
name_cn	否	String	站点信息自定义中文名称。 最小长度： <b>1</b> 最大长度： <b>64</b>
site_code	否	String	站点编码。 最小长度： <b>1</b> 最大长度： <b>64</b>
site_type	否	String	站点类型： <ul style="list-style-type: none"><li>• Area: 大区</li><li>• SubArea: 区域</li><li>• Region: 城域</li></ul>

## 请求参数

表 4-502 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-503 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
site_infos	Array of GlobalConnectionBandwidthSites objects	站点信息列表响应体。 数组长度：1 - 10

表 4-504 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-505 GlobalConnectionBandwidthSites

参数	参数类型	描述
id	String	资源ID标识符。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
name_en	String	功能说明：站点信息自定义的英文名字。取值范围：1-255个字符 最小长度：1 最大长度：255
name_cn	String	功能说明：站点信息自定义的中文名字。取值范围：1-64个字符。 最小长度：1 最大长度：64
site_code	String	功能说明：站点编码，格式为<area_code>[-<subarea_code>[-<region_code>]]。取值范围：1-64个字符。 最小长度：1 最大长度：64
site_type	String	功能说明：站点类型，必须跟站点编码对应，一段编码为大区，两段编码为区域，三段编码为城域。取值范围： <ul style="list-style-type: none"><li>• Area: 大区站点</li><li>• SubArea: 区域站点</li><li>• Region: 城域站点</li></ul> 最小长度：1 最大长度：64
service_list	String	功能说明：站点支持的服务列表，多个服务用","分隔。取值范围：0-255个字符 最小长度：0 最大长度：255
group_list	Array of SiteGroupReferenceInfo objects	站点分组跟外部关联的数据模型。 数组长度：1 - 10

参数	参数类型	描述
region_id	String	功能说明：对应华为云标准region的id，该站点继承自华为云region时才需要填写该字段。取值范围：0-64个字符。 最小长度：0 最大长度：64
public_border_group	String	功能说明：用于标记是中心还是边缘站点。中心：center 取值范围：0-255个字符。 最小长度：0 最大长度：255

表 4-506 SiteGroupReferenceInfo

参数	参数类型	描述
id	String	资源ID标识符。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
name_en	String	功能说明：站点分组自定义的英文名字。取值范围：1-255个字符。 最小长度：1 最大长度：255
name_cn	String	功能说明：站点分组自定义的中文名字。取值范围：1-64个字符。 最小长度：1 最大长度：64

## 请求示例

查询站点列表

```
GET https://{cc_endpoint}/v3/{domain_id}/gcb/sites
```

## 响应示例

**状态码：200**

查询站点列表成功。

```
{  
  "request_id": "61126320a1802d5c6444f9d2d76526c2",  
  "site_infos": [{  
    "id": "5c2d5343-cde2-4a41-83ae-56b7f44d73b5",  
    "description": "description",
```



```
"name_cn" : "CN north beijing4",
"name_en" : "CN north beijing4",
"site_code" : "cn-north-beijing4",
"site_type" : "Region",
"service_list" : "CC,GEIP",
"region_id" : "cn-north-4",
"public_border_group" : "center",
"group_list" : [ {
  "id" : "a3bad420-33b8-4e26-9e9b-bdf67aa8e72b",
  "description" : "description"
} ],
"created_at" : "2024-01-24T08:26:41.914Z",
"updated_at" : "2024-01-24T08:26:41.914Z"
}],
"page_info" : {
  "next_marker" : "XXX",
  "previous_marker" : "XXX",
  "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListGlobalConnectionBandwidthSitesSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListGlobalConnectionBandwidthSitesRequest request = new
        ListGlobalConnectionBandwidthSitesRequest();
        request.withSiteCode("<site_code>");

        request.withSiteType(ListGlobalConnectionBandwidthSitesRequest.SiteTypeEnum.fromValue("<site_type>"));
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withId();
    }
}
```

```
try {
    ListGlobalConnectionBandwidthSitesResponse response =
client.listGlobalConnectionBandwidthSites(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListGlobalConnectionBandwidthSitesRequest()
        request.site_code = "<site_code>"
        request.site_type = "<site_type>"
        request.limit = <limit>
        request.marker = "<marker>"
        request.id =
        response = client.list_global_connection_bandwidth_sites(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v3/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListGlobalConnectionBandwidthSitesRequest{}
    siteCodeRequest := "<site_code>"
    request.SiteCode = &siteCodeRequest
    siteTypeRequest := model.GetListGlobalConnectionBandwidthSitesRequestSiteTypeEnum().<SITE_TYPE>
    request.SiteType = &siteTypeRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListGlobalConnectionBandwidthSites(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	查询站点列表成功。

## 错误码

请参见[错误码](#)。

## 4.13.12 查询线路等级列表

### 功能介绍

查询线路等级列表。分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/gcb/line-levels

表 4-507 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户ID。 最小长度：10 最大长度：32

表 4-508 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页返回的个数。取值范围：1~1000。 最小值：1 最大值：1000 缺省值：1000
marker	否	String	翻页信息，从上次API调用返回的翻页数据中获取，可填写前一页marker或者后一页marker，填入前一页previous_marker就向前翻页，后一页next_marker就向翻页。翻页过程中，查询条件不能修改，包括过滤条件，排序条件，limit。 最小长度：1 最大长度：4096
id	否	Array	根据id查询，可查询多个id。 数组长度：1 - 20
local_area	否	String	线路规格本端接入点编码信息。 最小长度：1 最大长度：64
remote_area	否	String	线路规格远端接入点编码信息。 最小长度：1 最大长度：64

参数	是否必选	参数类型	描述
levels	否	Array	带宽等级信息： <ul style="list-style-type: none"><li>• Pt: 铂金</li><li>• Ag: 银</li></ul> 数组长度：1 - 10

## 请求参数

表 4-509 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 最小长度：0 最大长度：2000000

## 响应参数

状态码：200

表 4-510 响应 Body 参数

参数	参数类型	描述
request_id	String	资源ID标识符。 最小长度：32 最大长度：36
page_info	PageInfo object	分页查询页的信息。
line_levels	Array of GlobalConnectionBandwidthLineLevel objects	线路分级列表响应体。 数组长度：1 - 2000

表 4-511 PageInfo

参数	参数类型	描述
next_marker	String	向后分页标识符。 最小长度：1 最大长度：4096

参数	参数类型	描述
previous_marker	String	向前分页标识符。 最小长度：1 最大长度：4096
current_count	Integer	当前列表中资源数量。 最小值：0 最大值：2000

表 4-512 GlobalConnectionBandwidthLineLevel

参数	参数类型	描述
id	String	资源ID标识符。
description	String	实例描述。不支持 <>。 最小长度：0 最大长度：255
created_at	String	实例创建时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
updated_at	String	实例更新时间。UTC时间格式，yyyy-MM-ddTHH:mm:ss。
local_area	String	功能说明：线路分级本端接入点。 最小长度：1 最大长度：64
remote_area	String	功能描述：线路分级远端接入点。 最小长度：1 最大长度：64
levels	Array of strings	支持的铂金、金、银分级。 最小长度：1 最大长度：36 数组长度：1 - 10

## 请求示例

查询线路等级列表

```
GET https://{cc_endpoint}/v3/{domain_id}/gcb/line-levels
```

## 响应示例

状态码：200

查询线路等级列表成功。

```
{
  "request_id": "61126320a1802d5c6444f9d2d76526c2",
  "line_levels": [ {
    "id": "5c2d5343-cde2-4a41-83ae-56b7f44d73b5",
    "description": "description",
    "local_area": "cn-north-beijing4",
    "remote_area": "cn-south-guangzhou",
    "levels": [ "Pt", "Ag" ],
    "created_at": "2024-01-24T08:26:41.914Z",
    "updated_at": "2024-01-24T08:26:41.914Z"
  } ],
  "page_info": {
    "next_marker": "XXX",
    "previous_marker": "XXX",
    "current_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.cc.v3.region.CcRegion;
import com.huaweicloud.sdk.cc.v3.*;
import com.huaweicloud.sdk.cc.v3.model.*;

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

public class ListGlobalConnectionBandwidthLineLevelsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        ListGlobalConnectionBandwidthLineLevelsRequest request = new
        ListGlobalConnectionBandwidthLineLevelsRequest();
        request.withLocalArea("<local_area>");
        request.withRemoteArea("<remote_area>");
        request.withLevels();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withId();
        try {
```

```
ListGlobalConnectionBandwidthLineLevelsResponse response =
client.listGlobalConnectionBandwidthLineLevels(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdkcc.v3.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListGlobalConnectionBandwidthLineLevelsRequest()
        request.local_area = "<local_area>"
        request.remote_area = "<remote_area>"
        request.levels =
        request.limit = <limit>
        request.marker = "<marker>"
        request.id =
        response = client.list_global_connection_bandwidth_line_levels(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

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



```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListGlobalConnectionBandwidthLineLevelsRequest{}
    localAreaRequest:= "<local_area>"
    request.LocalArea = &localAreaRequest
    remoteAreaRequest:= "<remote_area>"
    request.RemoteArea = &remoteAreaRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListGlobalConnectionBandwidthLineLevels(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	查询线路等级列表成功。

## 错误码

请参见[错误码](#)。

## 4.14 全域互联带宽标签管理

## 4.14.1 查询账户全域互联带宽资源的标签

### 功能介绍

查询账户全域互联带宽资源的标签

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/gcb/{resource\_id}/tags

表 4-513 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64

### 请求参数

表 4-514 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

### 响应参数

状态码：200

表 4-515 响应 Body 参数

参数	参数类型	描述
tags	Array of <b>Tag</b> objects	标签列表。 数组长度：0 - 20
request_id	String	请求ID。 最小长度：0 最大长度：36

表 4-516 Tag

参数	参数类型	描述
key	String	键 最小长度：1 最大长度：128
value	String	值 最小长度：0 最大长度：255

## 请求示例

查询账户全域互联带宽资源标签

```
GET https://{cc_endpoint}/v3/gcb/{resource_id}/tags
```

## 响应示例

状态码：200

查询账户全域互联带宽资源标签成功。

```
{  
  "tags": [{  
    "key": "DEV",  
    "value": "DEV1"  
  }],  
  "request_id": "d6bf8eeb1f2b6e503e88f3dc3024c7e7"  
}
```

## 状态码

状态码	描述
200	查询账户全域互联带宽资源标签成功。

## 错误码

请参见[错误码](#)。

## 4.14.2 添加账户全域互联带宽资源标签

### 功能介绍

添加账户全域互联带宽资源标签

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/gcb/{resource\_id}/tags

表 4-517 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64

## 请求参数

表 4-518 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

表 4-519 请求 Body 参数

参数	是否必选	参数类型	描述
tag	是	<b>RequiredTag</b> object	一个key/value键值对

表 4-520 RequiredTag

参数	是否必选	参数类型	描述
key	是	String	键 最小长度：1 最大长度：128
value	是	String	值 最小长度：0 最大长度：255

## 响应参数

无

## 请求示例

添加账户全域互联带宽资源标签

```
POST https://{cc_endpoint}/v3/gcb/{resource_id}/tags
{
  "tag" : {
    "key" : "DEV",
    "value" : "DEV1"
  }
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

添加账户全域互联带宽资源标签

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

public class CreateGcbResourceTagSolution {

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

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

        CreateGcbResourceTagRequest request = new CreateGcbResourceTagRequest();
        CreateGcbTagRequestBody body = new CreateGcbTagRequestBody();
        RequiredTag tagbody = new RequiredTag();
        tagbody.withKey("DEV")
            .withValue("DEV1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateGcbResourceTagResponse response = client.createGcbResourceTag(request);
            System.out.println(response.toString());
        }
    }
}
```

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

## Python

### 添加账户全域互联带宽资源标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateGcbResourceTagRequest()
        tagbody = RequiredTag(
            key="DEV",
            value="DEV1"
        )
        request.body = CreateGcbTagRequestBody(
            tag=tagbody
        )
        response = client.create_gcb_resource_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 添加账户全域互联带宽资源标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateGcbResourceTagRequest{}
    tagbody := &model.RequiredTag{
        Key: "DEV",
        Value: "DEV1",
    }
    request.Body = &model.CreateGcbTagRequestBody{
        Tag: tagbody,
    }
    response, err := client.CreateGcbResourceTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，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](#)。

### 4.14.3 删除账户全域互联带宽资源标签

#### 功能介绍

删除账户全域互联带宽资源标签

## 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/gcb/{resource\_id}/tags/{tag\_key}

表 4-521 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64
tag_key	是	String	删除的tag的key。 最小长度：0 最大长度：128

## 请求参数

表 4-522 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token 最小长度：0 最大长度：102400

## 响应参数

无

## 请求示例

删除账户全域互联带宽资源标签

```
DELETE https://{cc_endpoint}/v3/gcb/{resource_id}/tags/{tag_key}
```

## 响应示例

无



## 状态码

状态码	描述
204	操作成功。

## 错误码

请参见[错误码](#)。

### 4.14.4 批量添加账户全域互联带宽资源标签

#### 功能介绍

TMS批量添加资源标签

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/gcb/{resource\_id}/tags/create

表 4-523 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64

#### 请求参数

表 4-524 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

表 4-525 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>RequiredTag</b> objects	批量添加/删除资源标签。 数组长度： <b>0 - 20</b>

表 4-526 RequiredTag

参数	是否必选	参数类型	描述
key	是	String	键 最小长度： <b>1</b> 最大长度： <b>128</b>
value	是	String	值 最小长度： <b>0</b> 最大长度： <b>255</b>

## 响应参数

无

## 请求示例

批量添加账户全域互联带宽资源标签

```
POST https://{cc_endpoint}/v3/gcb/{resource_id}/tags/create
```

```
{
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value2"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

批量添加账户全域互联带宽资源标签

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

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

public class BatchCreateGcbResourceTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateGcbResourceTagsRequest request = new BatchCreateGcbResourceTagsRequest();
        CreateDeleteGcbTagsRequestBody body = new CreateDeleteGcbTagsRequestBody();
        List<RequiredTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new RequiredTag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new RequiredTag()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateGcbResourceTagsResponse response = client.batchCreateGcbResourceTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 批量添加账户全域互联带宽资源标签

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateGcbResourceTagsRequest()
        listTagsbody = [
            RequiredTag(
                key="key1",
                value="value1"
            ),
            RequiredTag(
                key="key2",
                value="value2"
            )
        ]
        request.body = CreateDeleteGcbTagsRequestBody(
            tags=listTagsbody
        )
        response = client.batch_create_gcb_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

### 批量添加账户全域互联带宽资源标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchCreateGcbResourceTagsRequest{}
var listTagsbody = []model.RequiredTag{
    {
        Key: "key1",
        Value: "value1",
    },
    {
        Key: "key2",
        Value: "value2",
    },
}
request.Body = &model.CreateDeleteGcbTagsRequestBody{
    Tags: listTagsbody,
}
response, err := client.BatchCreateGcbResourceTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，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](#)。

## 4.14.5 批量删除账户全域互联带宽资源标签

### 功能介绍

批量删除账户全域互联带宽资源标签

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v3/gcb/{resource\_id}/tags/delete

表 4-527 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64

## 请求参数

表 4-528 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

表 4-529 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>RequiredTag</b> objects	批量添加/删除资源标签。 数组长度：0 - 20

表 4-530 RequiredTag

参数	是否必选	参数类型	描述
key	是	String	键 最小长度：1 最大长度：128
value	是	String	值 最小长度：0 最大长度：255

## 响应参数

无

## 请求示例

### 批量删除账户全域互联带宽资源标签

```
DELETE https://{cc_endpoint}/v3/gcb/{resource_id}/tags/delete

{
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

#### 批量删除账户全域互联带宽资源标签

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

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

public class BatchDeleteGcbResourceTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteGcbResourceTagsRequest request = new BatchDeleteGcbResourceTagsRequest();
        CreateDeleteGcbTagsRequestBody body = new CreateDeleteGcbTagsRequestBody();
        List<RequiredTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new RequiredTag()
```

```
        .withKey("key1")
        .withValue("value1")
    );
    listbodyTags.add(
        new RequiredTag()
            .withKey("key2")
            .withValue("value2")
    );
    body.withTags(listbodyTags);
    request.withBody(body);
    try {
        BatchDeleteGcbResourceTagsResponse response = client.batchDeleteGcbResourceTags(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

### 批量删除账户全域互联带宽资源标签

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v2 import *
```

```
if __name__ == "__main__":
```

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

```
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
            .with_credentials(credentials) \
            .with_region(CcRegion.value_of("<YOUR REGION>")) \
            .build()
```

```
    try:
```

```
        request = BatchDeleteGcbResourceTagsRequest()
```

```
        listTagsbody = [
```

```
            RequiredTag(
                key="key1",
                value="value1"
```

```
            ),
```

```
            RequiredTag(
                key="key2",
                value="value2"
```

```
        )
```

```
    ]
```

```
    request.body = CreateDeleteGcbTagsRequestBody(
        tags=listTagsbody
    )
```



```
response = client.batch_delete_gcb_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

### 批量删除账户全域互联带宽资源标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteGcbResourceTagsRequest{}
    var listTagsbody = []model.RequiredTag{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    request.Body = &model.CreateDeleteGcbTagsRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.BatchDeleteGcbResourceTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，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](#)。

## 4.14.6 查询账户全域互联带宽资源标签数量

### 功能介绍

查询账户全域互联带宽资源标签数量

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/gcb/resource-instances/count

### 请求参数

表 4-531 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

表 4-532 请求 Body 参数

参数	是否必选	参数类型	描述
request_id	否	String	请求ID。 最小长度：0 最大长度：36
without_any_tag	否	Boolean	不包含任意一个标签，该字段为true时查询所有不带标签的资源，此时忽略“tags”、“tags_any”、“not_tags”、“not_tags_any”字段。

参数	是否必选	参数类型	描述
tags	否	Array of <a href="#">QueryTag</a> objects	包含标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>
tags_any	否	Array of <a href="#">QueryTag</a> objects	包含任意标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>
not_tags	否	Array of <a href="#">QueryTag</a> objects	不包含标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回不包含标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>
not_tags_any	否	Array of <a href="#">QueryTag</a> objects	不包含任意标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回不包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>

参数	是否必选	参数类型	描述
matches	否	Array of <a href="#">TmsMatch</a> objects	是否匹配以下tag, key必须为 "resource_name", value如果有值则模糊匹配, 如果为空字符串则精确匹配。 数组长度: <b>0 - 50</b>

表 4-533 QueryTag

参数	是否必选	参数类型	描述
key	是	String	键。 最小长度: <b>0</b> 最大长度: <b>128</b>
values	是	Array of strings	值列表。 最小长度: <b>0</b> 最大长度: <b>255</b> 数组长度: <b>1 - 10</b>

表 4-534 TmsMatch

参数	是否必选	参数类型	描述
key	是	String	键。第一期限定为 resource_name, 后续扩展。 最小长度: <b>1</b> 最大长度: <b>128</b>
value	是	String	值。 最小长度: <b>0</b> 最大长度: <b>255</b>

## 响应参数

状态码: 200

表 4-535 响应 Body 参数

参数	参数类型	描述
total_count	Integer	总记录数。 最小值：0 最大值：2000.0
request_id	String	请求ID。 最小长度：0 最大长度：36

## 请求示例

查询账户全域互联带宽资源标签数量

POST https://{cc\_endpoint}/v3/gcb/resource-instances/count

```
{
  "not_tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "tags_any": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "not_tags_any": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "matches": [ {
    "key": "resource_name",
    "value": "resource1"
  } ]
}
```

## 响应示例

**状态码： 200**

查询账户全域互联带宽资源标签数量。

```
{
  "request_id": "e3371202-7e85-4c16-bb4b-4e3c092734f7",
  "total_count": 1
}
```

## 状态码

状态码	描述
200	查询账户全域互联带宽资源标签数量。

## 错误码

请参见[错误码](#)。

## 4.14.7 查询账户全域互联带宽资源实例列表

### 功能介绍

查询账户全域互联带宽资源实例列表

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/gcb/resource-instances/filter

表 4-536 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询记录数。 最小值：1 最大值：1000.0
offset	否	Integer	索引位置，偏移量。 最小值：0 最大值：999

### 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-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

表 4-538 请求 Body 参数

参数	是否必选	参数类型	描述
request_id	否	String	请求ID。 最小长度：0 最大长度：36
without_any_tag	否	Boolean	不包含任意一个标签，该字段为true时查询所有不带标签的资源，此时忽略“tags”、“tags_any”、“not_tags”、“not_tags_any”字段。
tags	否	Array of QueryTag objects	包含标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 数组长度：0 - 50
tags_any	否	Array of QueryTag objects	包含任意标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：0 - 50
not_tags	否	Array of QueryTag objects	不包含标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回不包含标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：0 - 50

参数	是否必选	参数类型	描述
not_tags_any	否	Array of <a href="#">QueryTag</a> objects	不包含任意标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回不包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>
matches	否	Array of <a href="#">TmsMatch</a> objects	是否匹配以下tag，key必须为"resource_name"，value如果有值则模糊匹配，如果为空字符串则精确匹配。 数组长度： <b>0 - 50</b>

表 4-539 QueryTag

参数	是否必选	参数类型	描述
key	是	String	键。 最小长度： <b>0</b> 最大长度： <b>128</b>
values	是	Array of strings	值列表。 最小长度： <b>0</b> 最大长度： <b>255</b> 数组长度： <b>1 - 10</b>

表 4-540 TmsMatch

参数	是否必选	参数类型	描述
key	是	String	键。第一期限定为resource_name,后续扩展。 最小长度： <b>1</b> 最大长度： <b>128</b>
value	是	String	值。 最小长度： <b>0</b> 最大长度： <b>255</b>



## 响应参数

状态码： 200

表 4-541 响应 Body 参数

参数	参数类型	描述
resources	Array of <b>TmsResource</b> objects	数组长度： 0 - 2000
total_count	Integer	总记录数。 最小值： 0 最大值： 2000.0
request_id	String	Request ID 最小长度： 0 最大长度： 36

表 4-542 TmsResource

参数	参数类型	描述
resource_id	String	资源ID。 最小长度： 1 最大长度： 36
tags	Array of <b>Tag</b> objects	标签列表，没有标签默认为空数组。 数组长度： 0 - 20
resource_name	String	资源名称，资源没有名称时默认为空字符串。 最小长度： 1 最大长度： 64

表 4-543 Tag

参数	参数类型	描述
key	String	键 最小长度： 1 最大长度： 128
value	String	值 最小长度： 0 最大长度： 255

## 请求示例

### 查询账户全域互联带宽资源实例列表

POST https://{cc\_endpoint}/v3/gcb/resource-instances/filter

```
{
  "matches": [ {
    "key": "resource_name",
    "value": "resource1"
  } ],
  "not_tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "tags_any": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "not_tags_any": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ]
}
```

## 响应示例

### 状态码： 200

查询账户全域互联带宽资源实例列表。

```
{
  "resources": [ {
    "resource_id": "cb9d7ead-834a-4275-93e1-217f79df9dc1",
    "resource_name": "test",
    "tags": [ {
      "key": "key1",
      "value": "value1"
    } ]
  } ],
  "request_id": "e3371202-7e85-4c16-bb4b-4e3c092734f7",
  "total_count": 1
}
```

## 状态码

状态码	描述
200	查询账户全域互联带宽资源实例列表。

## 错误码

请参见[错误码](#)。

## 4.14.8 查询账户全域互联带宽所有资源标签

### 功能介绍

查询账户全域互联带宽所有资源标签

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/gcb/tags

### 请求参数

表 4-544 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

### 响应参数

状态码：200

表 4-545 响应 Body 参数

参数	参数类型	描述
tags	Array of <a href="#">TmsTagValue</a> s objects	标签列表。 数组长度：0 - 20
total_count	Integer	总记录数。 最小值：0 最大值：2000.0
request_id	String	请求ID。 最小长度：0 最大长度：36

表 4-546 TmsTagValues

参数	参数类型	描述
key	String	键。 最小长度：0 最大长度：128
values	Array of strings	值列表。 最小长度：0 最大长度：255 数组长度：1 - 10

## 请求示例

查询账户全域互联带宽所有资源标签

```
GET https://{cc_endpoint}/v3/gcb/tags
```

## 响应示例

**状态码：200**

查询账户全域互联带宽所有资源标签。

```
{
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "request_id": "e3371202-7e85-4c16-bb4b-4e3c092734f7",
  "total_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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

public class ListGcbTenantTagsSolution {

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

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListGcbTenantTagsRequest request = new ListGcbTenantTagsRequest();
try {
    ListGcbTenantTagsResponse response = client.listGcbTenantTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdccc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListGcbTenantTagsRequest()
        response = client.list_gcb_tenant_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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询账户全域互联带宽所有资源标签。

## 错误码

请参见[错误码](#)。

# 5 权限与授权项

## 5.1 权限与授权项说明

如果您需要对您所拥有的Cloud Connect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），如果华为账号所具备的权限功能已经能满足您的要求，您可以跳过本章节，不影响您使用Cloud Connect服务的其它功能。

通过IAM，您可以通过授权控制主体（IAM用户、用户组、IAM委托）对华为云资源的访问范围。目前IAM支持两类授权，一类是角色与策略授权，另一类为身份策略授权。

两者有如下的区别和关系：

表 5-1 两类授权的区别

名称	核心关系	涉及的权限	授权方式	适用场景
角色与策略授权	用户-权限-授权范围	<ul style="list-style-type: none"><li>系统角色</li><li>系统策略</li><li>自定义策略</li></ul>	为主体授予角色或策略	核心关系为“用户-权限-授权范围”，每个用户根据所需权限和所需授权范围进行授权，无法直接给用户授权，需要维护更多的用户组，且支持的条件键较少，难以满足细粒度精确权限控制需求，更适用于对细粒度权限管控要求较低的中小企业用户。

名称	核心关系	涉及的权限	授权方式	适用场景
身份策略授权	用户-策略	<ul style="list-style-type: none"><li>系统身份策略</li><li>自定义身份策略</li></ul>	<ul style="list-style-type: none"><li>为主体授予身份策略</li><li>身份策略附加至主体</li></ul>	核心关系为“用户-策略”，管理员可根据业务需求定制不同的访问控制策略，能够做到更细粒度更灵活的权限控制，新增资源时，对比角色与策略授权，基于身份策略的授权模型可以更快地直接给用户授权，灵活性更强，更方便，但相对应的，整体权限管控模型构建更加复杂，对相关人员专业能力要求更高，因此更适用于中大型企业。

例如：如果需要对IAM用户授予可以创建华北-北京四区域的ECS和华南-广州区域的OBS的权限，基于角色与策略授权的场景中，管理员需要创建两个自定义策略，并且为IAM用户同时授予这两个自定义策略才可以实现权限控制。在基于身份策略授权的场景中，管理员仅需要创建一个自定义策略，在策略中通过条件键

“g:RequestedRegion”的配置即可达到策略对于授权区域的控制。将策略附加主体或为主体授予该策略即可获得相应权限，权限配置方式更细粒度更灵活。

两种授权场景下的策略/身份策略、授权项等并不互通，推荐使用身份策略进行授权。

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云连接列表，那么在基于角色与策略授权的场景中，这个IAM用户被授予的权限中必须包含允许“cc:cloudConnections:list”的授权项，该接口才能调用成功。在基于身份策略授权的场景中，这个IAM用户被授予的权限中包含“cc:cloudConnections:list”的授权项，该接口才能调用成功。

## 5.2 策略授权参考

本章节介绍Cloud Connect策略授权场景下支持的策略授权项。

### 支持的授权项

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

- 权限：允许或拒绝对指定资源在特定条件下进行某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。
- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。



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

CC的支持自定义策略授权项如下所示：

- **云连接**，包含所有云连接接口对应的授权项，如创建云连接、更新云连接、删除云连接、查询云连接详情、查询云连接列表等接口。
- **网络实例**，包含所有网络实例接口对应的授权项，如创建网络实例、更新网络实例、删除网络实例、查询网络实例详情、查询网络实例列表等接口。
- **带宽包**，包含所有带宽包接口对应的授权项，如创建带宽包、更新带宽包、删除带宽包、查询带宽包详情、查询带宽包列表、关联带宽包、解关联带宽包等接口。
- **域间带宽**，包含所有域间带宽接口对应的授权项，如创建域间带宽、更新域间带宽、删除域间带宽、查询域间带宽详情、查询域间带宽列表等接口。
- **云连接路由**，包含所有云连接路由接口对应的授权项，如查询云连接路由详情、查询云连接路由等接口。
- **中心网络**，包含所有中心网络接口对应的授权项，如创建中心网络、更新中心网络、删除中心网络、查询中心网络详情、查询中心网络列表等接口。
- **中心网络策略**，包含所有中心网络策略接口对应的授权项，如创建中心网络策略、应用中心网络策略、删除中心网络策略、查询中心网络策略列表、查询策略变化集等接口。
- **中心网络连接**，包含所有中心网络连接接口对应的授权项，如查询中心网络连接列表、更新中心网络连接等接口。
- **中心网络附件**，包含所有中心网络附件接口对应的授权项，如创建中心网络GDGW附件、更新中心网络GDGW附件、查询中心网络附件详情、查询中心网络GDGW附件列表、删除中心网络附件、查询中心网络附件列表等接口。

## 云连接

表 5-2 云连接

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建云连接	POST /v3/{domain_id}/ccaas/cloud-connections	cc:cloudConnections:create	-	√	√

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
更新云连接	PUT /v3/{domain_id}/ccaas/cloud-connections/{id}	cc:cloudConnections:update	-	√	√
删除云连接	DELETE /v3/{domain_id}/ccaas/cloud-connections/{id}	cc:cloudConnections:delete	-	√	√
查询云连接详情	GET /v3/{domain_id}/ccaas/cloud-connections/{id}	cc:cloudConnections:get	-	√	√
查询云连接列表	GET /v3/{domain_id}/ccaas/cloud-connections	cc:cloudConnections:list	-	√	√

## 网络实例

表 5-3 网络实例

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建网络实例	POST /v3/{domain_id}/ccaas/network-instances	cc:networkInstances:create	-	√	×
更新网络实例	PUT /v3/{domain_id}/ccaas/network-instances/{id}	cc:networkInstances:update	-	√	×
删除网络实例	DELETE /v3/{domain_id}/ccaas/network-instances/{id}	cc:networkInstances:delete	-	√	×

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询网络实例详情	GET /v3/{domain_id}/ccaas/network-instances/{id}	cc:networkInstances:get	-	√	×
查询网络实例列表	GET /v3/{domain_id}/ccaas/network-instances	cc:networkInstances:list	-	√	×

## 带宽包

表 5-4 带宽包

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建带宽包	POST /v3/{domain_id}/ccaas/bandwidth-packages	cc:bandwidthPackages:create	-	√	√
更新带宽包	PUT /v3/{domain_id}/ccaas/bandwidth-packages/{id}	cc:bandwidthPackages:update	-	√	√
删除带宽包	DELETE /v3/{domain_id}/ccaas/bandwidth-packages/{id}	cc:bandwidthPackages:delete	-	√	√
查询带宽包详情	GET /v3/{domain_id}/ccaas/bandwidth-packages/{id}	cc:bandwidthPackages:get	-	√	√

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询带宽包列表	GET /v3/{domain_id}/ccaas/bandwidth-packages	cc:bandwidthPackages:list	-	√	√
关联带宽包	POST /v3/{domain_id}/ccaas/bandwidth-packages/{id}/associate	cc:bandwidthPackages:associate	-	√	√
解关联带宽包	POST /v3/{domain_id}/ccaas/bandwidth-packages/{id}/disassociate	cc:bandwidthPackages:disassociate	-	√	√

## 域间带宽

表 5-5 域间带宽

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建域间带宽	POST /v3/{domain_id}/ccaas/inter-region-bandwidths	cc:interRegionBandwidths:create	-	√	×
更新域间带宽	PUT /v3/{domain_id}/ccaas/inter-region-bandwidths/{id}	cc:interRegionBandwidths:update	-	√	×

权限	对应API接口	授权项 ( Action )	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
删除域间带宽	DELETE /v3/{domain_id}/ccaas/inter-region-bandwidths/{id}	cc:interRegionBandwidths:delete	-	√	×
查询域间带宽详情	GET /v3/{domain_id}/ccaas/inter-region-bandwidths/{id}	cc:interRegionBandwidths:get	-	√	×
查询域间带宽列表	GET /v3/{domain_id}/ccaas/inter-region-bandwidths	cc:interRegionBandwidths:list	-	√	×

## 云连接路由

表 5-6 云连接路由

权限	对应API接口	授权项 ( Action )	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询云连接路由详情	GET /v3/{domain_id}/ccaas/cloud-connection-routes/{id}	cc:cloudConnectionRoutes:get	-	√	×
查询云连接路由	GET /v3/{domain_id}/ccaas/cloud-connection-routes	cc:cloudConnectionRoutes:list	-	√	×

## 中心网络

表 5-7 中心网络

权限	对应API接口	授权项 ( Action )	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建中心网络	POST /v3/{domain_id}/gcn/central-networks	cc:central Network:create	-	√	√
更新中心网络	PUT /v3/{domain_id}/gcn/central-networks/{central_network_id}	cc:central Network:update	-	√	√
删除中心网络	DELETE /v3/{domain_id}/gcn/central-networks/{central_network_id}	cc:central Network:delete	-	√	√
查询中心网络详情	GET /v3/{domain_id}/gcn/central-networks/{central_network_id}	cc:central Network:get	-	√	√
查询中心网络列表	GET /v3/{domain_id}/gcn/central-networks	cc:central Network:list	-	√	√

## 中心网络策略

表 5-8 中心网络策略

权限	对应API接口	授权项 ( Action )	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建中心网络策略	POST /v3/{domain_id}/gcn/central-network/{central_network_id}/policies	cc:centralNetwork:createPolicy	-	√	√
应用中心网络策略	POST /v3/{domain_id}/gcn/central-network/{central_network_id}/policies/{policy_id}/apply	cc:centralNetwork:applyPolicy	-	√	√
删除中心网络策略	DELETE /v3/{domain_id}/gcn/central-network/{central_network_id}/policies/{policy_id}	cc:centralNetwork:deletePolicy	-	√	√
查询中心网络策略列表	GET /v3/{domain_id}/gcn/central-network/{central_network_id}/policies	cc:centralNetwork:listPolicies	-	√	√
查询策略变化集	GET /v3/{domain_id}/gcn/central-network/{central_network_id}/policies/{policy_id}/change-set	cc:centralNetwork:listChangeSet	-	√	√

## 中心网络连接

表 5-9 中心网络连接

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询中心网络连接列表	GET /v3/{domain_id}/gcn/central-network/{central_network_id}/connections	cc:centralNetwork:listConnections	-	√	√
更新中心网络连接	PUT /v3/{domain_id}/gcn/central-network/{central_network_id}/connections/{connection_id}	cc:centralNetwork:updateConnection	-	√	√

## 中心网络附件

表 5-10 中心网络附件

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建中心网络GDGW附件	POST /v3/{domain_id}/gcn/central-network/{central_network_id}/gdgw-attachments	cc:centralNetworkAttachment:createGdgw	-	√	√
更新中心网络GDGW附件	PUT /v3/{domain_id}/gcn/central-network/{central_network_id}/gdgw-attachments/{gdgw_attachment_id}	cc:centralNetworkAttachment:updateGdgw	-	√	√



权限	对应API接口	授权项 ( Action )	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询中心网络附件详情	GET /v3/{domain_id}/gcn/central-network/{central_network_id}/gdgw-attachments/{gdgw_attachment_id}	cc:centralNetworkAttachment:getGdgw	-	√	√
查询中心网络GDGW附件列表	GET /v3/{domain_id}/gcn/central-network/{central_network_id}/gdgw-attachments	cc:centralNetworkAttachment:listGdgws	-	√	√
删除中心网络附件	DELETE /v3/{domain_id}/gcn/central-network/{central_network_id}/attachments/{attachment_id}	cc:centralNetworkAttachment:delete	-	√	√
查询中心网络附件列表	GET /v3/{domain_id}/gcn/central-network/{central_network_id}/attachments	cc:centralNetworkAttachment:list	-	√	√

# 6 附录

## 6.1 状态码

- 正常

表 6-1 正常时的返回值

状态码	返回值	状态码说明
200	OK	POST、GET和PUT操作正常返回。
201	Created	POST操作正常返回。
204	No Content	DELETE操作正常返回。

- 异常

表 6-2 异常时的返回值

状态码	返回值	状态码说明
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。
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。

状态码	返回值	状态码说明
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。

## 6.2 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。

状态码	错误码	错误信息	描述	处理措施
400	CC.1001	The parameter is invalid.	入参非法。	请输入合法的参数。
400	CC.1006	Tenants are restricted and cannot operate APIs such as adding or modifying resources.	账户受限。	请联系技术支持。
400	CC.1007	Internal Server Error.	调用API出现异常。	请联系技术支持。
400	CC.1031	The cloud connection name already in use.	云连接名称重复。	请更换云连接名称。

状态码	错误码	错误信息	描述	处理措施
400	CC.1032	The cloud connection cannot be deleted because it has network instances.	云连接下有实例，不能删除。	请先移除云连接下网络实例再执行删除云连接操作。
400	CC.1033	The cloud connection quota is insufficient.	云连接配额不足。	请联系技术支持。
400	CC.1041	Internal Server Error.	内部未知异常。	请联系技术支持。
400	CC.1042	The cloud connection instance is frozen.	云连接被冻结。	请联系技术支持。
400	CC.1043	ER type cloud connection is not supported.	不支持ER类型云连接。	请联系技术支持。
400	CC.1050	Please wait for approval.	跨境申请未通过。	请检查跨境申请审批状态。
400	CC.1051	No application for cross-border.	未提交跨境申请。	请先提交跨境申请。
400	CC.1052	The bandwidth package used in regionBandwidth cannot be deleted.	带宽包被域间带宽使用。	请先删除域间带宽，再删除带宽包。
400	CC.1053	The bandwidth package name is already in use.	带宽包名称不唯一。	请更换带宽包名称。
400	CC.1054	The bandwidth package is frozen.	冻结的带宽包不允许操作。	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	CC.1055	The bandwidth package has been bound.	带宽包已经被绑定。	请检查输入的带宽包信息。
400	CC.1056	Used RegionBandwidth is bigger than new BwpSize.	带宽值不能小于被使用的带宽。	请检查带宽包已经被使用的带宽值。
400	CC.1057	The account does not have the permission to operation.	未加入白名单。	请联系技术支持。
400	CC.1061	Failed to delete the bandwidth package persistence.	带宽包删除返回值错误。	请联系技术支持。
400	CC.1062	Failed to update the bandwidth package persistence.	带宽包更新返回值错误。	请联系技术支持。
400	CC.1070	Same-direction instance is existed in cloud connection.	云连接中存在同一方向实例。	请检查云连接中网络实例信息。
400	CC.1071	Region Bandwidth Size is greater than bandwidth package size.	域间带宽大小大于带宽包大小。	请调小域间带宽值，或者升配带宽包。
400	CC.1072	ER type cloud connection does not support 95-billed bandwidth packages.	ER类型云连接不支持95计费带宽包。	ER类型云连接请绑定其他类型的带宽包。
400	CC.1081	Internal Server Error.	内部未知异常。	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	CC.1090	The instance has been used.	网络实例已经被加载。	请使用其他网络实例进行加载。
400	CC.1091	The CIDR of a network instance under the same cloud connection is unique.	同一个云连接下的网络实例CIDR唯一。	请使用不同的CIDR。
400	CC.1092	The instance is not authorized.	跨账号场景下网络实例未授权不允许加载。	请检查网络实例跨账号授权情况。
400	CC.1093	The network instance routing conflict.	前VPC存在路由冲突，此VPC路由已经在对等连接、VPN等其它场景中使用。	请检查路由冲突。
400	CC.1094	The ER instance status is not available.	ER实例状态有异常。	请检查ER实例状态。
400	CC.1095	The ER attachment status not available.	加载ER实例时，重复创建了Attachment。	请联系技术支持。
400	CC.1096	The ER attachment quota is insufficient.	加载ER实例时，Attachment配额不足。	请联系技术支持。
400	CC.1110	Cross account authorization does not support cloud connection instances of this account.	跨账号授权不支持本账号云连接实例。	请检查授权的云连接实例信息。
400	CC.1111	The network instance has been authorized.	网络实例已被授权。	请检查授权的网络实例信息。

状态码	错误码	错误信息	描述	处理措施
400	CC.1171	Related bandwidth package is not available.	相关带宽包不可用。	请联系技术支持。
400	CC.1250	If an enterprise project has been enabled you need to specify the enterprise project when creating resources.	企业项目为空。	请在创建资源时指定企业项目。
400	CC.1269	To create a resource in an enterprise project,you need to apply it first.	企业项目不为空。	请不要在创建资源时指定企业项目。
400	CC.1290	The cloud connection quota is insufficient.	云连接实例配额不足。	请联系技术支持。
400	CC.1291	No quotas for instance for this region.	云连接实例下单Region可加载网络实例配额不足。	请联系技术支持。
400	CC.1292	No quotas for region for this cloudconnect.	云连接实例下可加载Region配额不足。	请联系技术支持。
400	CC.1293	No quotas for route for this cloudconnect.	云连接实例下可加载路由配额不足。	请联系技术支持。
400	CC.1380	The resource processing, please try again later.	处理中, 请稍后重试。	请稍后重试。

状态码	错误码	错误信息	描述	处理措施
400	CC.1390	Domain Common Data Consent unable to operate non-Chinese mainland region.	您还未签署云连接服务声明，无法操作非中国大陆Region的资源。	请前往云连接控制台列表页面进行签署。
401	CC.1003	The parameter of domainId is illegal.	账户非法。	请检查租户ID的合法性。
403	CC.1004	You do not have permission to operate.	禁止访问。	请联系技术支持。
404	CC.1002	The resource could not be found.	未找到资源。	请输入正确的资源信息。
409	CC.1005	The resource already exists.	资源已存在。	请勿重复创建。
500	CC.1000	Internal Server Error.	内部未知异常。	请联系技术支持。
500	CC.1251	Internal Server Error.	内部未知异常。	请联系技术支持。

## 6.3 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 6-1 查看项目 ID



# 7 历史 API

## 7.1 标签管理

### 7.1.1 查询账户资源标签

#### 功能介绍

查询账户资源标签

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{domain\_id}/ccaas/{resource\_type}/tags

表 7-1 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID 最小长度：10 最大长度：32
resource_type	是	String	资源类型： <ul style="list-style-type: none"><li>cloud-connection: 云连接</li><li>bandwidth-package: 带宽包</li></ul>

## 请求参数

表 7-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token 最小长度：0 最大长度：102400

## 响应参数

状态码：200

表 7-3 响应 Body 参数

参数	参数类型	描述
tags	Array of <a href="#">AggTag</a> objects	标签列表 数组长度：0 - 20
request_id	String	请求ID。 最小长度：0 最大长度：36

表 7-4 AggTag

参数	参数类型	描述
key	String	键 最小长度：1 最大长度：128
values	Array of strings	相同键的值列表 最小长度：0 最大长度：255 数组长度：1 - 10

## 请求示例

查询账户资源标签

```
GET https://{cc_endpoint}/v3/{domain_id}/ccaas/{resource_type}/tags
```

## 响应示例

状态码： 200

查询账户资源标签成功

```
{
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  }, {
    "key": "key2",
    "values": [ "value1", "value2" ]
  } ]
}
```

## 状态码

状态码	描述
200	查询账户资源标签成功

## 错误码

请参见[错误码](#)。

## 7.1.2 删除资源标签

### 功能介绍

删除资源标签

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{domain\_id}/ccaas/{resource\_type}/{resource\_id}/tags/{tag\_key}

表 7-5 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID 最小长度：10 最大长度：32
resource_id	是	String	资源ID 最小长度：0 最大长度：36

参数	是否必选	参数类型	描述
tag_key	是	String	待删除资源标签的key 最小长度：1 最大长度：128
resource_type	是	String	资源类型： <ul style="list-style-type: none"><li>cloud-connection: 云连接</li><li>bandwidth-package: 带宽包</li></ul>

## 请求参数

表 7-6 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token 最小长度：0 最大长度：102400

## 响应参数

无

## 请求示例

删除资源标签

```
DELETE https://{cc_endpoint}/v3/{domain_id}/ccaas/{resource_type}/{resource_id}/tags/{tag_key}
```

## 响应示例

无

## 状态码

状态码	描述
204	删除标签成功

## 错误码

请参见[错误码](#)。

## 7.1.3 批量创建和删除资源标签

### 功能介绍

批量创建和删除标签

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/{resource\_type}/{resource\_id}/tags/action

表 7-7 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID 最小长度：10 最大长度：32
resource_id	是	String	资源ID 最小长度：0 最大长度：36
resource_type	是	String	资源类型： <ul style="list-style-type: none"><li>cloud-connection: 云连接</li><li>bandwidth-package: 带宽包</li></ul>

### 请求参数

表 7-8 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token 最小长度：0 最大长度：102400

表 7-9 请求 Body 参数

参数	是否必选	参数类型	描述
action	否	String	动作。 - create: 创建。 delete: 删除。

参数	是否必选	参数类型	描述
tags	否	Array of <b>Tag</b> objects	批量添加/删除资源标签 数组长度： <b>0 - 20</b>

表 7-10 Tag

参数	是否必选	参数类型	描述
key	否	String	键 最小长度： <b>1</b> 最大长度： <b>128</b>
value	否	String	值 最小长度： <b>0</b> 最大长度： <b>255</b>

## 响应参数

无

## 请求示例

- 批量创建资源标签，如果key相同则更新value，如果key不存在则添加value  
POST `https://{cc_endpoint}/v3/{domain_id}/ccaas/{resource_type}/{resource_id}/tags/action`

```
{
  "action": "create",
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value3"
    }
  ]
}
```

- 批量删除资源标签，仅根据key进行删除，忽略value  
POST `https://{cc_endpoint}/v3/{domain_id}/ccaas/{resource_type}/{resource_id}/tags/action`

```
{
  "action": "delete",
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value3"
    }
  ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

- 批量创建资源标签，如果key相同则更新value，如果key不存在则添加value

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

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

public class BatchCreateDeleteTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateDeleteTagsRequest request = new BatchCreateDeleteTagsRequest();
        ResourceTags body = new ResourceTags();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value3")
        );
        body.withTags(listbodyTags);
        body.withAction(ResourceTags.ActionEnum.fromValue("create"));
        request.withBody(body);
        try {
            BatchCreateDeleteTagsResponse response = client.batchCreateDeleteTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
        }
    }
}
```



```
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- 批量删除资源标签，仅根据key进行删除，忽略value

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

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

public class BatchCreateDeleteTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateDeleteTagsRequest request = new BatchCreateDeleteTagsRequest();
        ResourceTags body = new ResourceTags();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value3")
        );
        body.withTags(listbodyTags);
        body.withAction(ResourceTags.ActionEnum.fromValue("delete"));
        request.withBody(body);
        try {
            BatchCreateDeleteTagsResponse response = client.batchCreateDeleteTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

## Python

- 批量创建资源标签，如果key相同则更新value，如果key不存在则添加value

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkcc.v2.region.cc_region import CcRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcc.v2 import *
```

```
if __name__ == "__main__":
```

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

```
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CcRegion.value_of("<YOUR REGION>")) \  
        .build()
```

```
try:
```

```
    request = BatchCreateDeleteTagsRequest()  
    listTagsbody = [  
        Tag(  
            key="key1",  
            value="value1"  
        ),  
        Tag(  
            key="key2",  
            value="value3"  
        )  
    ]
```

```
    request.body = ResourceTags(  
        tags=listTagsbody,  
        action="create"  
    )
```

```
    response = client.batch_create_delete_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)
```

- 批量删除资源标签，仅根据key进行删除，忽略value

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials  
from huaweicloudsdkcc.v2.region.cc_region import CcRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcc.v2 import *
```

```
if __name__ == "__main__":
```

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

```
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
.with_credentials(credentials) \
.with_region(CcRegion.value_of("<YOUR REGION>")) \
.build()

try:
request = BatchCreateDeleteTagsRequest()
listTagsbody = [
    Tag(
        key="key1",
        value="value1"
    ),
    Tag(
        key="key2",
        value="value3"
    )
]
request.body = ResourceTags(
    tags=listTagsbody,
    action="delete"
)
response = client.batch_create_delete_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

- 批量创建资源标签，如果key相同则更新value，如果key不存在则添加value

```
package main

import (
    "fmt"
    "github.com/ HuaweiCloud/ huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/ HuaweiCloud/ huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/ HuaweiCloud/ huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/ HuaweiCloud/ huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.BatchCreateDeleteTagsRequest{}
keyTags:= "key1"
valueTags:= "value1"
keyTags1:= "key2"
valueTags1:= "value3"
var listTagsbody = []model.Tag{
    {
        Key: &keyTags,
        Value: &valueTags,
    },
    {
        Key: &keyTags1,
        Value: &valueTags1,
    },
}
actionResourceTags:= model.GetResourceTagsActionEnum().CREATE
request.Body = &model.ResourceTags{
    Tags: &listTagsbody,
    Action: &actionResourceTags,
}
response, err := client.BatchCreateDeleteTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 批量删除资源标签，仅根据key进行删除，忽略value

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateDeleteTagsRequest{}
    keyTags:= "key1"
    valueTags:= "value1"
    keyTags1:= "key2"
    valueTags1:= "value3"
    var listTagsbody = []model.Tag{
        {
            Key: &keyTags,
            Value: &valueTags,
        },
    },
}
```

```
{
    Key: &keyTags1,
    Value: &valueTags1,
},
}
actionResourceTags:= model.GetResourceTagsActionEnum().DELETE
request.Body = &model.ResourceTags{
    Tags: &listTagsbody,
    Action: &actionResourceTags,
}
response, err := client.BatchCreateDeleteTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
204	批量创建和删除标签成功

## 错误码

请参见[错误码](#)。

## 7.1.4 查询资源实例

### 功能介绍

查询资源实例

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/{resource\_type}/resource-instances/action

表 7-11 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最小长度：10 最大长度：32

参数	是否必选	参数类型	描述
resource_type	是	String	资源类型: <ul style="list-style-type: none"><li>cloud-connection: 云连接</li><li>bandwidth-package: 带宽包</li></ul>

## 请求参数

表 7-12 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token 最小长度: <b>0</b> 最大长度: <b>102400</b>

表 7-13 请求 Body 参数

参数	是否必选	参数类型	描述
action	否	String	动作。 - filter: 过滤。count: 查询总条数。
limit	否	Integer	查询结果数量限制 最小值: <b>1</b> 最大值: <b>1000.0</b>
offset	否	Integer	查询结果偏移 最小值: <b>0</b> 最大值: <b>999</b>
tags	否	Array of <b>AggTag</b> objects	是否包含以下tag (多个key取"与"关系, 多个value取"或"关系) 数组长度: <b>0 - 50</b>
matches	否	Array of <b>Tag</b> objects	是否匹配以下tag, key必须为"resource_name", value如果有值则模糊匹配, 如果为空字符串则精确匹配 数组长度: <b>0 - 50</b>

表 7-14 AggTag

参数	是否必选	参数类型	描述
key	否	String	键 最小长度：1 最大长度：128
values	否	Array of strings	相同键的值列表 最小长度：0 最大长度：255 数组长度：1 - 10

表 7-15 Tag

参数	是否必选	参数类型	描述
key	否	String	键 最小长度：1 最大长度：128
value	否	String	值 最小长度：0 最大长度：255

## 响应参数

状态码：200

表 7-16 响应 Body 参数

参数	参数类型	描述
resources	Array of <a href="#">FilterTagResource</a> objects	资源实例列表。 数组长度：0 - 2000
request_id	String	请求ID。 最小长度：0 最大长度：36
total_count	Integer	符合过滤条件的资源总数量。 最小值：0 最大值：2000.0

表 7-17 FilterTagResource

参数	参数类型	描述
resource_id	String	资源ID 最小长度：0 最大长度：36
resource_name	String	资源名称 最小长度：0 最大长度：64
resource_detail	String	资源详情 最小长度：0 最大长度：255
tags	Array of Tag objects	资源下包含的标签 数组长度：0 - 20

表 7-18 Tag

参数	参数类型	描述
key	String	键 最小长度：1 最大长度：128
value	String	值 最小长度：0 最大长度：255

## 请求示例

- 根据指定筛选条件查询资源例详情

POST https://{cc\_endpoint}/v3/{domain\_id}/ccaas/{resource\_type}/resource-instances/action

```
{
  "offset": "0",
  "limit": "100",
  "action": "filter",
  "matches": [ {
    "key": "resource_name",
    "value": "resource1"
  } ],
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ]
}
```

- 根据指定筛选条件查询资源实例数量

POST https://{cc\_endpoint}/v3/{domain\_id}/ccaas/cloud-connection/resource-instances/action



```
{
  "action": "count",
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  }, {
    "key": "key2",
    "values": [ "value1", "value2" ]
  } ],
  "matches": [ {
    "key": "resource_name",
    "value": "resource1"
  } ]
}
```

## 响应示例

**状态码： 200**

查询资源实例成功

```
{
  "resources": [ {
    "resource_id": "cdfs_cefs_wesas_12_dsad",
    "resource_name": "resoucee1",
    "tags": [ {
      "key": "key1",
      "value": "value1"
    }, {
      "key": "key2",
      "value": "value1"
    } ]
  } ],
  "request_id": "XXX",
  "total_count": 1000
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 根据指定筛选条件查询资源例详情

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

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

public class ListResourceByFilterTagSolution {

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListResourceByFilterTagRequest request = new ListResourceByFilterTagRequest();
ListResourceByFilterTagRequestBody body = new ListResourceByFilterTagRequestBody();
List<Tag> listbodyMatches = new ArrayList<>();
listbodyMatches.add(
    new Tag()
        .withKey("resource_name")
        .withValue("resource1")
);
List<String> listTagsValues = new ArrayList<>();
listTagsValues.add("value1");
listTagsValues.add("value2");
List<AggTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new AggTag()
        .withKey("key1")
        .withValues(listTagsValues)
);
body.withMatches(listbodyMatches);
body.withTags(listbodyTags);
body.withOffset(0);
body.withLimit(100);
body.withAction(ListResourceByFilterTagRequestBody.ActionEnum.fromValue("filter"));
request.withBody(body);
try {
    ListResourceByFilterTagResponse response = client.listResourceByFilterTag(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 根据指定筛选条件查询资源实例数量

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

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

public class ListResourceByFilterTagSolution {

    public static void main(String[] args) {
```

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

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListResourceByFilterTagRequest request = new ListResourceByFilterTagRequest();
ListResourceByFilterTagRequestBody body = new ListResourceByFilterTagRequestBody();
List<Tag> listbodyMatches = new ArrayList<>();
listbodyMatches.add(
    new Tag()
        .withKey("resource_name")
        .withValue("resource1")
);
List<String> listTagsValues = new ArrayList<>();
listTagsValues.add("value1");
listTagsValues.add("value2");
List<String> listTagsValues1 = new ArrayList<>();
listTagsValues1.add("value1");
listTagsValues1.add("value2");
List<AggTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new AggTag()
        .withKey("key1")
        .withValues(listTagsValues1)
);
listbodyTags.add(
    new AggTag()
        .withKey("key2")
        .withValues(listTagsValues)
);
body.withMatches(listbodyMatches);
body.withTags(listbodyTags);
body.withAction(ListResourceByFilterTagRequestBody.ActionEnum.fromValue("count"));
request.withBody(body);
try {
    ListResourceByFilterTagResponse response = client.listResourceByFilterTag(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

- 根据指定筛选条件查询资源例详情

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import GlobalCredentials
```

```
from huaweicloudsdkcc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListResourceByFilterTagRequest()
        listMatchesbody = [
            Tag(
                key="resource_name",
                value="resource1"
            )
        ]
        listValuesTags = [
            "value1",
            "value2"
        ]
        listTagsbody = [
            AggTag(
                key="key1",
                values=listValuesTags
            )
        ]
        request.body = ListResourceByFilterTagRequestBody(
            matches=listMatchesbody,
            tags=listTagsbody,
            offset=0,
            limit=100,
            action="filter"
        )
        response = client.list_resource_by_filter_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 根据指定筛选条件查询资源实例数量

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListResourceByFilterTagRequest()
    listMatchesbody = [
        Tag(
            key="resource_name",
            value="resource1"
        )
    ]
    listValuesTags = [
        "value1",
        "value2"
    ]
    listValuesTags1 = [
        "value1",
        "value2"
    ]
    listTagsbody = [
        AggTag(
            key="key1",
            values=listValuesTags1
        ),
        AggTag(
            key="key2",
            values=listValuesTags
        )
    ]
    request.body = ListResourceByFilterTagRequestBody(
        matches=listMatchesbody,
        tags=listTagsbody,
        action="count"
    )
    response = client.list_resource_by_filter_tag(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

- 根据指定筛选条件查询资源例详情

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListResourceByFilterTagRequest{
    keyMatches:= "resource_name"
    valueMatches:= "resource1"
    var listMatchesbody = []model.Tag{
        {
            Key: &keyMatches,
            Value: &valueMatches,
        },
    }
    var listValuesTags = []string{
        "value1",
        "value2",
    }
    keyTags:= "key1"
    var listTagsbody = []model.AggTag{
        {
            Key: &keyTags,
            Values: &listValuesTags,
        },
    }
    offsetListResourceByFilterTagRequestBody:= int32(0)
    limitListResourceByFilterTagRequestBody:= int32(100)
    actionListResourceByFilterTagRequestBody:=
model.GetListResourceByFilterTagRequestBodyActionEnum().FILTER
    request.Body = &model.ListResourceByFilterTagRequestBody{
        Matches: &listMatchesbody,
        Tags: &listTagsbody,
        Offset: &offsetListResourceByFilterTagRequestBody,
        Limit: &limitListResourceByFilterTagRequestBody,
        Action: &actionListResourceByFilterTagRequestBody,
    }
    response, err := client.ListResourceByFilterTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 根据指定筛选条件查询资源实例数量

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment 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 := cc.NewCcClient(
    cc.CcClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListResourceByFilterTagRequest{
    keyMatches:= "resource_name"
    valueMatches:= "resource1"
    var listMatchesbody = []model.Tag{
        {
            Key: &keyMatches,
            Value: &valueMatches,
        },
    }
    var listValuesTags = []string{
        "value1",
        "value2",
    }
    var listValuesTags1 = []string{
        "value1",
        "value2",
    }
    keyTags:= "key1"
    keyTags1:= "key2"
    var listTagsbody = []model.AggTag{
        {
            Key: &keyTags,
            Values: &listValuesTags1,
        },
        {
            Key: &keyTags1,
            Values: &listValuesTags,
        },
    }
    actionListResourceByFilterTagRequestBody:=
model.GetListResourceByFilterTagRequestBodyActionEnum().COUNT
    request.Body = &model.ListResourceByFilterTagRequestBody{
        Matches: &listMatchesbody,
        Tags: &listTagsbody,
        Action: &actionListResourceByFilterTagRequestBody,
    }
    response, err := client.ListResourceByFilterTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	查询资源实例成功

## 错误码

请参见[错误码](#)。

## 7.1.5 查询资源标签

### 功能介绍

查询资源标签

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{domain\_id}/ccaas/{resource\_type}/{resource\_id}/tags

表 7-19 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID 最小长度：10 最大长度：32
resource_type	是	String	资源类型： <ul style="list-style-type: none"><li>cloud-connection: 云连接</li><li>bandwidth-package: 带宽包</li></ul>
resource_id	是	String	资源ID 最小长度：0 最大长度：36



## 请求参数

表 7-20 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token 最小长度：0 最大长度：102400

## 响应参数

状态码：200

表 7-21 响应 Body 参数

参数	参数类型	描述
tags	Array of <b>Tag</b> objects	标签列表 数组长度：0 - 20
request_id	String	请求ID。 最小长度：0 最大长度：36

表 7-22 Tag

参数	参数类型	描述
key	String	键 最小长度：1 最大长度：128
value	String	值 最小长度：0 最大长度：255

## 请求示例

查询资源标签列表

GET https://{cc\_endpoint}/v3/{domain\_id}/ccaas/{resource\_type}/{resource\_id}/tags

## 响应示例

状态码：200

### 查询标签成功

```
{
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value3"
  } ]
}
```

### 状态码

状态码	描述
200	查询标签成功

### 错误码

请参见[错误码](#)。

## 7.1.6 添加资源标签

### 功能介绍

添加资源标签

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/ccaas/{resource\_type}/{resource\_id}/tags

表 7-23 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账号ID。 最小长度：10 最大长度：32
resource_id	是	String	资源ID。 最小长度：0 最大长度：36
resource_type	是	String	资源类型： <ul style="list-style-type: none"><li>cloud-connection: 云连接</li><li>bandwidth-package: 带宽包</li></ul>

## 请求参数

表 7-24 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token 最小长度：0 最大长度：102400

表 7-25 请求 Body 参数

参数	是否必选	参数类型	描述
tag	否	Tag object	一个key/value键值对

表 7-26 Tag

参数	是否必选	参数类型	描述
key	否	String	键 最小长度：1 最大长度：128
value	否	String	值 最小长度：0 最大长度：255

## 响应参数

无

## 请求示例

添加资源标签

POST https://{cc\_endpoint}/v3/{domain\_id}/ccaas/{resource\_type}/{resource\_id}/tags

```
{
  "tag": {
    "key": "key1",
    "value": "value1"
  }
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

#### 添加资源标签

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

public class CreateTagSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateTagRequest request = new CreateTagRequest();
        CreateTagRequestBody body = new CreateTagRequestBody();
        Tag tagbody = new Tag();
        tagbody.withKey("key1")
            .withValue("value1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateTagResponse response = client.createTag(request);
            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

#### 添加资源标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdccc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateTagRequest()
        tagbody = Tag(
            key="key1",
            value="value1"
        )
        request.body = CreateTagRequestBody(
            tag=tagbody
        )
        response = client.create_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 添加资源标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

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

```
cc.CcClientBuilder().
  WithRegion(region.ValueOf("<YOUR REGION>")).
  WithCredential(auth).
  Build())

request := &model.CreateTagRequest{
  keyTag:= "key1"
  valueTag:= "value1"
  tagbody := &model.Tag{
    Key: &keyTag,
    Value: &valueTag,
  }
  request.Body = &model.CreateTagRequestBody{
    Tag: tagbody,
  }
}
response, err := client.CreateTag(request)
if err == nil {
  fmt.Printf("%+v\n", response)
} else {
  fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
204	创建标签成功

## 错误码

请参见[错误码](#)。

## 7.2 全域互联带宽标签管理

### 7.2.1 查询账户全域互联带宽资源的标签

#### 功能介绍

查询账户全域互联带宽资源的标签

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/gcb/{resource\_id}/tags

表 7-27 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64

## 请求参数

表 7-28 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

## 响应参数

状态码：200

表 7-29 响应 Body 参数

参数	参数类型	描述
tags	Array of <b>Tag</b> objects	标签列表。 数组长度：0 - 20
request_id	String	请求ID。 最小长度：0 最大长度：36

表 7-30 Tag

参数	参数类型	描述
key	String	键 最小长度：1 最大长度：128
value	String	值 最小长度：0 最大长度：255

## 请求示例

查询账户全域互联带宽资源标签

```
GET https://{cc_endpoint}/v3/gcb/{resource_id}/tags
```

## 响应示例

**状态码： 200**

查询账户全域互联带宽资源标签成功。

```
{
  "tags": [ {
    "key": "DEV",
    "value": "DEV1"
  } ],
  "request_id": "d6bf8eeb1f2b6e503e88f3dc3024c7e7"
}
```

## 状态码

状态码	描述
200	查询账户全域互联带宽资源标签成功。

## 错误码

请参见[错误码](#)。

## 7.2.2 添加账户全域互联带宽资源标签

### 功能介绍

添加账户全域互联带宽资源标签

### 调用方法

请参见[如何调用API](#)。

### URI

```
POST /v3/gcb/{resource_id}/tags
```

表 7-31 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64



## 请求参数

表 7-32 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>102400</b>

表 7-33 请求 Body 参数

参数	是否必选	参数类型	描述
tag	是	<b>RequiredTag</b> object	一个key/value键值对

表 7-34 RequiredTag

参数	是否必选	参数类型	描述
key	是	String	键 最小长度： <b>1</b> 最大长度： <b>128</b>
value	是	String	值 最小长度： <b>0</b> 最大长度： <b>255</b>

## 响应参数

无

## 请求示例

添加账户全域互联带宽资源标签

```
POST https://{cc_endpoint}/v3/gcb/{resource_id}/tags
```

```
{  
  "tag": {  
    "key": "DEV",  
    "value": "DEV1"  
  }  
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

添加账户全域互联带宽资源标签

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

public class CreateGcbResourceTagSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateGcbResourceTagRequest request = new CreateGcbResourceTagRequest();
        CreateGcbTagRequestBody body = new CreateGcbTagRequestBody();
        RequiredTag tagbody = new RequiredTag();
        tagbody.withKey("DEV")
            .withValue("DEV1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateGcbResourceTagResponse response = client.createGcbResourceTag(request);
            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

添加账户全域互联带宽资源标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateGcbResourceTagRequest()
        tagbody = RequiredTag(
            key="DEV",
            value="DEV1"
        )
        request.body = CreateGcbTagRequestBody(
            tag=tagbody
        )
        response = client.create_gcb_resource_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 添加账户全域互联带宽资源标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

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

```
cc.CcClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build()

request := &model.CreateGcbResourceTagRequest{}
tagbody := &model.RequiredTag{
    Key: "DEV",
    Value: "DEV1",
}
request.Body = &model.CreateGcbTagRequestBody{
    Tag: tagbody,
}
response, err := client.CreateGcbResourceTag(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，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](#)。

## 7.2.3 删除账户全域互联带宽资源标签

### 功能介绍

删除账户全域互联带宽资源标签

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/gcb/{resource\_id}/tags/{tag\_key}

表 7-35 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64
tag_key	是	String	删除的tag的key。 最小长度：0 最大长度：128

## 请求参数

表 7-36 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token 最小长度：0 最大长度：102400

## 响应参数

无

## 请求示例

删除账户全域互联带宽资源标签

```
DELETE https://{cc_endpoint}/v3/gcb/{resource_id}/tags/{tag_key}
```

## 响应示例

无

## 状态码

状态码	描述
204	操作成功。

## 错误码

请参见[错误码](#)。

## 7.2.4 批量添加账户全域互联带宽资源标签

### 功能介绍

TMS批量添加资源标签

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/gcb/{resource\_id}/tags/create

表 7-37 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64

### 请求参数

表 7-38 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

表 7-39 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>RequiredTag</b> objects	批量添加/删除资源标签。 数组长度：0 - 20

表 7-40 RequiredTag

参数	是否必选	参数类型	描述
key	是	String	键 最小长度：1 最大长度：128
value	是	String	值 最小长度：0 最大长度：255

## 响应参数

无

## 请求示例

批量添加账户全域互联带宽资源标签

```
POST https://{cc_endpoint}/v3/gcb/{resource_id}/tags/create
```

```
{
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value2"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

批量添加账户全域互联带宽资源标签

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

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

```
public class BatchCreateGcbResourceTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateGcbResourceTagsRequest request = new BatchCreateGcbResourceTagsRequest();
        CreateDeleteGcbTagsRequestBody body = new CreateDeleteGcbTagsRequestBody();
        List<RequiredTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new RequiredTag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new RequiredTag()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateGcbResourceTagsResponse response = client.batchCreateGcbResourceTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### 批量添加账户全域互联带宽资源标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CcRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = BatchCreateGcbResourceTagsRequest()
    listTagsbody = [
        RequiredTag(
            key="key1",
            value="value1"
        ),
        RequiredTag(
            key="key2",
            value="value2"
        )
    ]
    request.body = CreateDeleteGcbTagsRequestBody(
        tags=listTagsbody
    )
    response = client.batch_create_gcb_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

### 批量添加账户全域互联带宽资源标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateGcbResourceTagsRequest{}
    var listTagsbody = []model.RequiredTag{
```

```
{
  Key: "key1",
  Value: "value1",
},
{
  Key: "key2",
  Value: "value2",
},
}
request.Body = &model.CreateDeleteGcbTagsRequestBody{
  Tags: listTagsbody,
}
response, err := client.BatchCreateGcbResourceTags(request)
if err == nil {
  fmt.Printf("%v\n", response)
} else {
  fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，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](#)。

## 7.2.5 批量删除账户全域互联带宽资源标签

### 功能介绍

批量删除账户全域互联带宽资源标签

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/gcb/{resource\_id}/tags/delete

表 7-41 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源唯一标识符。 最小长度：0 最大长度：64

## 请求参数

表 7-42 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

表 7-43 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of <b>RequiredTag</b> objects	批量添加/删除资源标签。 数组长度：0 - 20

表 7-44 RequiredTag

参数	是否必选	参数类型	描述
key	是	String	键 最小长度：1 最大长度：128
value	是	String	值 最小长度：0 最大长度：255

## 响应参数

无

## 请求示例

批量删除账户全域互联带宽资源标签

```
DELETE https://{cc_endpoint}/v3/gcb/{resource_id}/tags/delete

{
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value2"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

#### 批量删除账户全域互联带宽资源标签

```
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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

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

public class BatchDeleteGcbResourceTagsSolution {

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

        CcClient client = CcClient.newBuilder()
            .withCredential(auth)
            .withRegion(CcRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteGcbResourceTagsRequest request = new BatchDeleteGcbResourceTagsRequest();
        CreateDeleteGcbTagsRequestBody body = new CreateDeleteGcbTagsRequestBody();
        List<RequiredTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new RequiredTag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
```

```
        new RequiredTag()
            .withKey("key2")
            .withValue("value2")
    );
    body.withTags(listbodyTags);
    request.withBody(body);
    try {
        BatchDeleteGcbResourceTagsResponse response = client.batchDeleteGcbResourceTags(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

### 批量删除账户全域互联带宽资源标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkcc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteGcbResourceTagsRequest()
        listTagsbody = [
            RequiredTag(
                key="key1",
                value="value1"
            ),
            RequiredTag(
                key="key2",
                value="value2"
            )
        ]
        request.body = CreateDeleteGcbTagsRequestBody(
            tags=listTagsbody
        )
        response = client.batch_delete_gcb_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

### 批量删除账户全域互联带宽资源标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteGcbResourceTagsRequest{}
    var listTagsbody = []model.RequiredTag{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    request.Body = &model.CreateDeleteGcbTagsRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.BatchDeleteGcbResourceTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，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](#)。

## 7.2.6 查询账户全域互联带宽资源标签数量

### 功能介绍

查询账户全域互联带宽资源标签数量

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/gcb/resource-instances/count

### 请求参数

表 7-45 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度： <b>0</b> 最大长度： <b>102400</b>

表 7-46 请求 Body 参数

参数	是否必选	参数类型	描述
request_id	否	String	请求ID。 最小长度： <b>0</b> 最大长度： <b>36</b>
without_any_tag	否	Boolean	不包含任意一个标签，该字段为true时查询所有不带标签的资源，此时忽略“tags”、“tags_any”、“not_tags”、“not_tags_any”字段。

参数	是否必选	参数类型	描述
tags	否	Array of <a href="#">QueryTag</a> objects	包含标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>
tags_any	否	Array of <a href="#">QueryTag</a> objects	包含任意标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>
not_tags	否	Array of <a href="#">QueryTag</a> objects	不包含标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回不包含标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>
not_tags_any	否	Array of <a href="#">QueryTag</a> objects	不包含任意标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回不包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>



参数	是否必选	参数类型	描述
matches	否	Array of <a href="#">TmsMatch</a> objects	是否匹配以下tag, key必须为 "resource_name", value如果有值则模糊匹配, 如果为空字符串则精确匹配。 数组长度: <b>0 - 50</b>

表 7-47 QueryTag

参数	是否必选	参数类型	描述
key	是	String	键。 最小长度: <b>0</b> 最大长度: <b>128</b>
values	是	Array of strings	值列表。 最小长度: <b>0</b> 最大长度: <b>255</b> 数组长度: <b>1 - 10</b>

表 7-48 TmsMatch

参数	是否必选	参数类型	描述
key	是	String	键。第一期限定为 resource_name, 后续扩展。 最小长度: <b>1</b> 最大长度: <b>128</b>
value	是	String	值。 最小长度: <b>0</b> 最大长度: <b>255</b>

## 响应参数

状态码: 200

表 7-49 响应 Body 参数

参数	参数类型	描述
total_count	Integer	总记录数。 最小值：0 最大值：2000.0
request_id	String	请求ID。 最小长度：0 最大长度：36

## 请求示例

查询账户全域互联带宽资源标签数量

POST https://{cc\_endpoint}/v3/gcb/resource-instances/count

```
{
  "not_tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "tags_any": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "not_tags_any": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "matches": [ {
    "key": "resource_name",
    "value": "resource1"
  } ]
}
```

## 响应示例

**状态码： 200**

查询账户全域互联带宽资源标签数量。

```
{
  "request_id": "e3371202-7e85-4c16-bb4b-4e3c092734f7",
  "total_count": 1
}
```

## 状态码

状态码	描述
200	查询账户全域互联带宽资源标签数量。

## 错误码

请参见[错误码](#)。

## 7.2.7 查询账户全域互联带宽资源实例列表

### 功能介绍

查询账户全域互联带宽资源实例列表

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/gcb/resource-instances/filter

表 7-50 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询记录数。 最小值：1 最大值：1000.0
offset	否	Integer	索引位置，偏移量。 最小值：0 最大值：999

### 请求参数

表 7-51 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

表 7-52 请求 Body 参数

参数	是否必选	参数类型	描述
request_id	否	String	请求ID。 最小长度：0 最大长度：36
without_any_tag	否	Boolean	不包含任意一个标签，该字段为true时查询所有不带标签的资源，此时忽略“tags”、“tags_any”、“not_tags”、“not_tags_any”字段。
tags	否	Array of QueryTag objects	包含标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 数组长度：0 - 50
tags_any	否	Array of QueryTag objects	包含任意标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：0 - 50
not_tags	否	Array of QueryTag objects	不包含标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回不包含标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：0 - 50

参数	是否必选	参数类型	描述
not_tags_any	否	Array of <a href="#">QueryTag</a> objects	不包含任意标签，最多包含20个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回不包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度： <b>0 - 50</b>
matches	否	Array of <a href="#">TmsMatch</a> objects	是否匹配以下tag，key必须为"resource_name"，value如果有值则模糊匹配，如果为空字符串则精确匹配。 数组长度： <b>0 - 50</b>

表 7-53 QueryTag

参数	是否必选	参数类型	描述
key	是	String	键。 最小长度： <b>0</b> 最大长度： <b>128</b>
values	是	Array of strings	值列表。 最小长度： <b>0</b> 最大长度： <b>255</b> 数组长度： <b>1 - 10</b>

表 7-54 TmsMatch

参数	是否必选	参数类型	描述
key	是	String	键。第一期限定为resource_name,后续扩展。 最小长度： <b>1</b> 最大长度： <b>128</b>
value	是	String	值。 最小长度： <b>0</b> 最大长度： <b>255</b>

## 响应参数

状态码： 200

表 7-55 响应 Body 参数

参数	参数类型	描述
resources	Array of <b>TmsResource</b> objects	数组长度： 0 - 2000
total_count	Integer	总记录数。 最小值： 0 最大值： 2000.0
request_id	String	Request ID 最小长度： 0 最大长度： 36

表 7-56 TmsResource

参数	参数类型	描述
resource_id	String	资源ID。 最小长度： 1 最大长度： 36
tags	Array of <b>Tag</b> objects	标签列表，没有标签默认为空数组。 数组长度： 0 - 20
resource_name	String	资源名称，资源没有名称时默认为空字符串。 最小长度： 1 最大长度： 64

表 7-57 Tag

参数	参数类型	描述
key	String	键 最小长度： 1 最大长度： 128
value	String	值 最小长度： 0 最大长度： 255

## 请求示例

### 查询账户全域互联带宽资源实例列表

POST https://{cc\_endpoint}/v3/gcb/resource-instances/filter

```
{
  "matches": [ {
    "key": "resource_name",
    "value": "resource1"
  } ],
  "not_tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "tags_any": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "not_tags_any": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ]
}
```

## 响应示例

**状态码： 200**

查询账户全域互联带宽资源实例列表。

```
{
  "resources": [ {
    "resource_id": "cb9d7ead-834a-4275-93e1-217f79df9dc1",
    "resource_name": "test",
    "tags": [ {
      "key": "key1",
      "value": "value1"
    } ]
  } ],
  "request_id": "e3371202-7e85-4c16-bb4b-4e3c092734f7",
  "total_count": 1
}
```

## 状态码

状态码	描述
200	查询账户全域互联带宽资源实例列表。

## 错误码

请参见[错误码](#)。

## 7.2.8 查询账户全域互联带宽所有资源标签

### 功能介绍

查询账户全域互联带宽所有资源标签

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/gcb/tags

### 请求参数

表 7-58 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 最小长度：0 最大长度：102400

### 响应参数

状态码：200

表 7-59 响应 Body 参数

参数	参数类型	描述
tags	Array of <a href="#">TmsTagValue</a> s objects	标签列表。 数组长度：0 - 20
total_count	Integer	总记录数。 最小值：0 最大值：2000.0
request_id	String	请求ID。 最小长度：0 最大长度：36



表 7-60 TmsTagValues

参数	参数类型	描述
key	String	键。 最小长度：0 最大长度：128
values	Array of strings	值列表。 最小长度：0 最大长度：255 数组长度：1 - 10

## 请求示例

查询账户全域互联带宽所有资源标签

```
GET https://{cc_endpoint}/v3/gcb/tags
```

## 响应示例

**状态码：200**

查询账户全域互联带宽所有资源标签。

```
{
  "tags": [ {
    "key": "key1",
    "values": [ "value1", "value2" ]
  } ],
  "request_id": "e3371202-7e85-4c16-bb4b-4e3c092734f7",
  "total_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.cc.v2.region.CcRegion;
import com.huaweicloud.sdk.cc.v2.*;
import com.huaweicloud.sdk.cc.v2.model.*;

public class ListGcbTenantTagsSolution {

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

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_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);

CcClient client = CcClient.newBuilder()
    .withCredential(auth)
    .withRegion(CcRegion.valueOf("<YOUR REGION>"))
    .build();
ListGcbTenantTagsRequest request = new ListGcbTenantTagsRequest();
try {
    ListGcbTenantTagsResponse response = client.listGcbTenantTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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 huaweicloudsdccc.v2.region.cc_region import CcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdccc.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored 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 = CcClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CcRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListGcbTenantTagsRequest()
        response = client.list_gcb_tenant_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"
    cc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cc/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment 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 := cc.NewCcClient(
        cc.CcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	查询账户全域互联带宽所有资源标签。

## 错误码

请参见[错误码](#)。

# A 修订记录

发布日期	修订记录
2024-03-06	第三次正式发布：增加《全域互联带宽》章节。
2022-10-23	第二次正式发布：增加《历史API》章节。
2022-05-30	第一次正式发布。