

标签管理服务

接口参考

文档版本 01
发布日期 2023-07-10



版权所有 © 华为技术有限公司 2024。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

安全声明

漏洞处理流程

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

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

如企业客户须获取漏洞信息，请参见如下网址：

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

目录

1 使用前必读	1
1.1 概述	1
1.2 调用说明	1
1.3 终端节点	1
1.4 约束与限制	1
1.5 基本概念	1
2 API 概览	3
3 如何调用 API	4
3.1 构造请求	4
3.2 认证鉴权	8
3.3 返回结果	9
4 快速入门	11
5 API 说明	13
5.1 查询版本操作	13
5.1.1 查询 API 版本列表	13
5.1.2 查询 API 版本号详情	21
5.2 预定义标签操作	31
5.2.1 创建预定义标签	31
5.2.2 删除预定义标签	41
5.2.3 查询预定义标签列表	51
5.2.4 修改预定义标签	63
5.3 配额	73
5.3.1 查询标签配额	73
5.4 资源标签	82
5.4.1 批量添加标签	82
5.4.2 批量移除标签	94
5.4.3 查询标签键列表	105
5.4.4 查询标签值列表	115
5.4.5 查询资源标签	125
5.4.6 根据标签过滤资源	134
5.5 查询标签管理支持的服务	148
5.5.1 查询标签管理支持的服务	148

6 权限策略与授权项	160
6.1 策略及授权项说明.....	160
6.2 标签管理服务 API 授权列表项.....	161
A 附录	163
A.1 标签支持的资源类型.....	163
A.2 状态码.....	163
A.3 错误码.....	164
A.4 获取项目 ID.....	166
B 修订记录	168

1 使用前必读

1.1 概述

欢迎使用标签管理服务（Tag Management Service，简称TMS）。标签用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。标签管理服务是一种快速便捷将标签集中管理的可视化服务，提供跨区域、跨服务的集中标签管理和资源分类功能。

您可以使用本文档提供API创建或删除预定义标签、查询预定义标签列表、修改预定义标签等。支持的全部操作请参见[API概览](#)。

在调用标签管理服务API之前，请确保已经充分了解标签管理服务相关概念，详细信息请参见[产品介绍](#)。

1.2 调用说明

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

1.3 终端节点

终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同，您可以从[地区和终端节点](#)中查询标签管理服务的终端节点。

1.4 约束与限制

- 您能创建的标签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“[配额调整](#)”。
- 更详细的限制请参见具体API的说明。

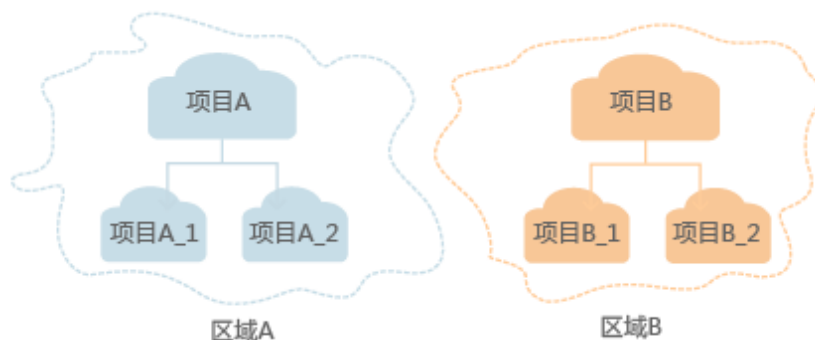
1.5 基本概念

- 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）。
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



- 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

表 2-1 TMS 接口列表

API	说明
查询API版本列表	查询TMS的API版本列表。
查询API版本号详情	查询TMS的API版本号详情。
创建预定义标签	创建预定义标签。用户创建预定义标签后，可以使用预定义标签来给资源添加标签。
删除预定义标签	删除预定义标签。用户可以删除已创建的预定义标签。
查询预定义标签列表	查询指定用户的预定义标签列表，访问数据库获取标签列表信息。
修改预定义标签	修改预定义标签。
查询标签配额	查询标签的配额信息。
批量添加标签	批量给云服务的多个资源添加标签。
批量移除标签	批量移除云服务多个资源的标签。
查询标签键列表	查询指定区域的所有标签键。
查询标签值列表	查询指定区域的标签键下的所有标签值。
查询资源标签	查询单个资源上的标签。
根据标签过滤资源	根据标签过滤资源。
查询标签管理支持的服务	查询标签管理服务支持的云服务。

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。

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。

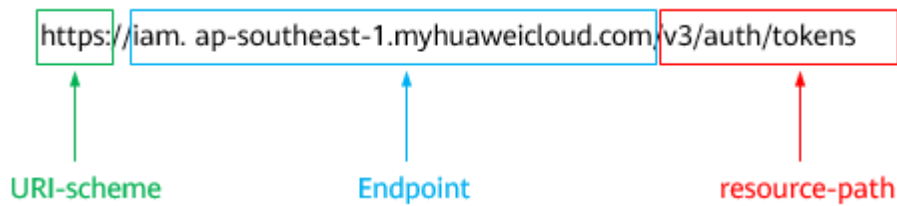
表 3-1 URI 中的参数说明

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 地区和终端节点 获取。 例如IAM服务在“中国-香港”区域的Endpoint为“iam.ap-southeast-1.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“中国-香港”区域的Token，则需使用“中国-香港”区域的Endpoint（iam.ap-southeast-1.myhuaweicloud.com），并在[获取用户Token](#)的URI部分找到resource-path（/v3/auth/tokens），拼接起来如下所示。

`https://iam.ap-southeast-1.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.ap-southeast-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 获取项目ID 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 获取用户Token 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.ap-southeast-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，***********为用户登录密码，***xxxxxxxxxxxxxxxxxxxx***为project的名称，如“ap-southeast-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.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

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

到这里为止这个请求需要的内容就具备齐全了，您可以使用[curl](#)、[Postman](#)或直接编写代码等方式发送请求调用API。对于获取用户Token接口，返回的响应消息头中的

“X-Subject-Token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

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

Token 认证

📖 说明

Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。

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

云服务存在两种部署方式：项目级服务和全局级服务。

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

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

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

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

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

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

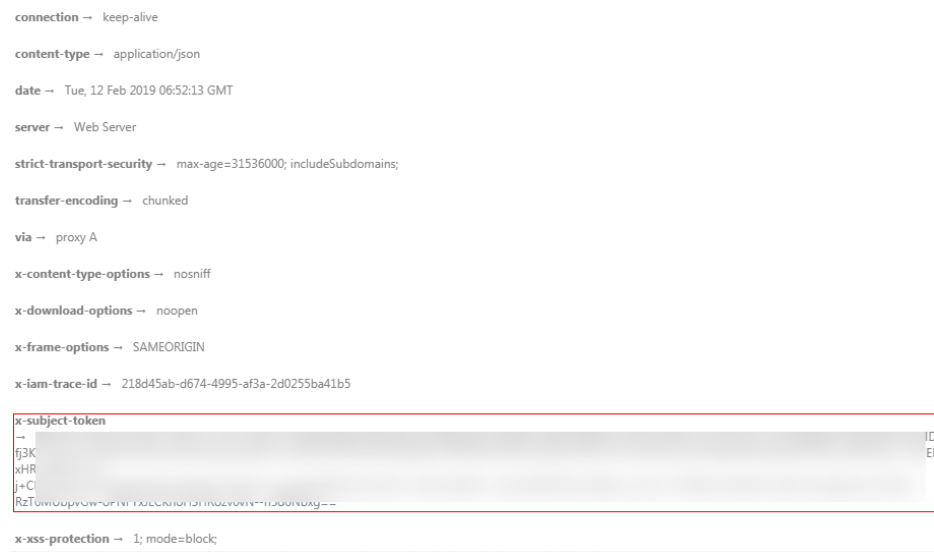
对应请求消息头，响应同样也有消息头，如“Content-type”。

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



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 快速入门

本节通过调用TMS的API创建预定义标签。

📖 说明

通过IAM服务获取到的Token有效期为24小时，需要使用同一个Token鉴权时，可以先将Token缓存，避免频繁调用。

涉及 API

当您使用Token认证方式完成认证鉴权时，需要获取用户Token并在调用接口时增加“X-Auth-Token”到业务接口请求消息头中。

- IAM获取token的API
- TMS创建预定义标签的API

具体步骤

1. Token认证，具体操作请参考[认证鉴权](#)。
 2. 发送“POST https://TMS的Endpoint/v1.0/predefine_tags/action”。
- 在Request Header中增加“Content-Type”和“X-Auth-Token”。

在Request Body中传入参数如下：

```
{
  "action": "create", //操作（必填，String）
  "tags": [
    {
      "key": "ENV1", //键（必填，String）
      "value": "DEV1" //值（必填，String）
    },
    {
      "key": "ENV2",
      "value": "DEV2"
    }
  ]
}
```

请求消息发送后，返回状态码204，响应体为空。

说明

- 响应消息包括成功和失败两类情况，本文以响应成功为例进行介绍。
- 若请求消息响应失败则会返回错误码及对应的错误信息说明，详细错误码信息请参见[错误码](#)。
- 响应消息的要素说明及返回值请参考“[创建预定义标签](#)”章节。

5 API 说明

5.1 查询版本操作

5.1.1 查询 API 版本列表

功能介绍

查询标签管理服务的API版本列表。

调用方法

请参见[如何调用API](#)。

URI

GET /

请求参数

无

响应参数

状态码： 200

表 5-1 响应 Body 参数

参数	参数类型	描述
versions	Array of VersionDetail objects	版本列表

表 5-2 VersionDetail

参数	参数类型	描述
id	String	版本ID（版本号），如v1.0。
links	Array of Link objects	API的URL地址。
version	String	若该版本API支持微版本，则返回支持的最新微版本号，如果不支持微版本，则返回空。
status	String	版本状态，为如下3种：CURRENT：表示该版本为主推版本。SUPPORTED：表示为老版本，但是现在还继续支持。DEPRECATED：表示为废弃版本，存在后续删除的可能。
updated	String	版本发布时间，采用UTC时间表示。如v1.0发布的时间2016-12-09T00:00:00Z。
min_version	String	若该版本API支持微版本，则返回支持的最早微版本号，如果不支持微版本，则返回空。

表 5-3 Link

参数	参数类型	描述
href	String	API的URL地址。
rel	String	self

状态码： 400

表 5-4 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-5 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-6 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-7 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-8 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-9 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-10 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-11 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-12 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-13 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-14 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-15 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-16 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-17 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-18 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-19 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-20 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-21 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-22 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-23 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-24 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-25 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

查询标签管理接口版本号列表

GET https://{Endpoint}/

响应示例

状态码： 200

OK

```
{
  "versions": [ {
    "id": "v1.0",
    "links": [ {
      "rel": "self",
      "href": "https://{Endpoint}/v1.0"
    } ],
    "version": "",
    "status": "CURRENT",
    "updated": "2016-12-09T00:00:00Z",
    "min_version": ""
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

public class ListApiVersionsSolution {

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

        TmsClient client = TmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListApiVersionsRequest request = new ListApiVersionsRequest();
        try {
            ListApiVersionsResponse response = client.listApiVersions(request);
            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 huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

    try:
        request = ListApiVersionsRequest()
        response = client.list_api_versions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

Go

```

package main

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

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

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

更多

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

状态码

状态码	描述
200	OK
400	Bad Request
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.1.2 查询 API 版本号详情

功能介绍

查询指定的标签管理服务API版本号详情。

调用方法

请参见[如何调用API](#)。

URI

GET /{api_version}

表 5-26 路径参数

参数	是否必选	参数类型	描述
api_version	是	String	API版本号。

请求参数

表 5-27 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

响应参数

状态码： 200

表 5-28 响应 Body 参数

参数	参数类型	描述
version	VersionDetail object	版本详情

表 5-29 VersionDetail

参数	参数类型	描述
id	String	版本ID（版本号），如v1.0。
links	Array of Link objects	API的URL地址。

参数	参数类型	描述
version	String	若该版本API支持微版本，则返回支持的最新微版本号，如果不支持微版本，则返回空。
status	String	版本状态，为如下3种：CURRENT：表示该版本为主推版本。SUPPORTED：表示为老版本，但是现在还继续支持。DEPRECATED：表示为废弃版本，存在后续删除的可能。
updated	String	版本发布时间，采用UTC时间表示。如v1.0发布的时间2016-12-09T00:00:00Z。
min_version	String	若该版本API支持微版本，则返回支持的最早微版本号，如果不支持微版本，则返回空。

表 5-30 Link

参数	参数类型	描述
href	String	API的URL地址。
rel	String	self

状态码： 400

表 5-31 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-32 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-33 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-34 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-35 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-36 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-37 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-38 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 408

表 5-39 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-40 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-41 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-42 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-43 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-44 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-45 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-46 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-47 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-48 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-49 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-50 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-51 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-52 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-53 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-54 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

查询标签管理接口版本号详情

```
GET https://{Endpoint}/v1.0
```

响应示例

状态码： 200

OK

```
{
  "version": {
    "id": "v1.0",
    "links": [ {
      "rel": "self",
      "href": "https://{Endpoint}/v1.0"
    } ],
    "version": "",
    "status": "CURRENT",
    "updated": "2016-12-09T00:00:00Z",
    "min_version": ""
  }
}
```

SDK 代码示例

SDK代码示例如下。

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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;
```

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

        TmsClient client = TmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowApiVersionRequest request = new ShowApiVersionRequest();
        try {
            ShowApiVersionResponse response = client.showApiVersion(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            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 huaweicloudsktms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsktms.v1 import *

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

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

```
print(e.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"
    tms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1/region"
)

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

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

更多

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

状态码

状态码	描述
200	OK
400	Bad Request
404	Not Found
405	Method Not Allowed
406	Not Acceptable

状态码	描述
408	Request Timeout
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.2 预定义标签操作

5.2.1 创建预定义标签

功能介绍

用于创建预定义标签。用户创建预定义标签后，可以使用预定义标签来给资源创建标签。该接口支持幂等特性和处理批量数据。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/predefine_tags/action

请求参数

表 5-55 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

表 5-56 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	操作标识（区分大小写）： create（创建）
tags	是	Array of PredefineTag Request objects	标签列表

表 5-57 PredefineTagRequest

参数	是否必选	参数类型	描述
key	是	String	键。最大长度36个字符。字符集：A-Z, a-z, 0-9, '-', '_', UNICOD字符（\u4E00-\u9FFF）。
value	是	String	值。每个值最大长度43个字符，可以为空字符串。字符集：A-Z, a-z, 0-9, '-', '_', UNICOD字符（\u4E00-\u9FFF）。

响应参数

状态码： 400

表 5-58 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-59 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-60 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-61 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-62 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-63 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-64 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-65 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-66 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-67 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-68 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-69 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-70 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-71 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-72 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-73 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-74 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-75 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-76 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-77 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-78 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-79 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-80 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-81 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-82 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-83 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

创建预定义标签

POST https://{Endpoint}/v1.0/predefine_tags/action

```
{
  "action": "create",
  "tags": [ {
    "key": "ENV1",
    "value": "DEV1"
  }, {
    "key": "ENV2",
    "value": "DEV2"
  } ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

创建预定义标签

```
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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

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

public class CreatePredefineTagsSolution {

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

TmsClient client = TmsClient.newBuilder()
    .withCredential(auth)
    .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
    .build();
CreatePredefineTagsRequest request = new CreatePredefineTagsRequest();
ReqCreatePredefineTag body = new ReqCreatePredefineTag();
List<PredefineTagRequest> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new PredefineTagRequest()
        .withKey("ENV1")
        .withValue("DEV1")
);
listbodyTags.add(
    new PredefineTagRequest()
        .withKey("ENV2")
        .withValue("DEV2")
);
body.withTags(listbodyTags);
body.withAction(ReqCreatePredefineTag.ActionEnum.fromValue("create"));
request.withBody(body);
try {
    CreatePredefineTagsResponse response = client.createPredefineTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

Python

创建预定义标签

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

```

```

try:
    request = CreatePredefineTagsRequest()
    listTagsbody = [
        PredefineTagRequest(
            key="ENV1",
            value="DEV1"
        ),
        PredefineTagRequest(
            key="ENV2",
            value="DEV2"
        )
    ]
    request.body = ReqCreatePredefineTag(
        tags=listTagsbody,
        action="create"
    )
    response = client.create_predefine_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

创建预定义标签

```

package main

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

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

    request := &model.CreatePredefineTagsRequest{}
    var listTagsbody = []model.PredefineTagRequest{
        {
            Key: "ENV1",
            Value: "DEV1",
        },
        {
            Key: "ENV2",
            Value: "DEV2",
        },
    }
}

```

```

request.Body = &model.ReqCreatePredefineTag{
    Tags: listTagsbody,
    Action: model.GetReqCreatePredefineTagActionEnum().CREATE,
}
response, err := client.CreatePredefineTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

更多

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

状态码

状态码	描述
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.2.2 删除预定义标签

功能介绍

用于删除预定义标签。该接口支持幂等特性和处理批量数据。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/predefine_tags/action

请求参数

表 5-84 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

表 5-85 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	操作标识（区分大小写）： delete（删除）
tags	是	Array of PredefineTag Request objects	标签列表，数量在1到50之间

表 5-86 PredefineTagRequest

参数	是否必选	参数类型	描述
key	是	String	键。最大长度36个字符。字符集：A-Z，a-z，0-9，‘-’，‘_’，UNICODE字符（\u4E00-\u9FFF）。
value	是	String	值。每个值最大长度43个字符，可以为空字符串。字符集：A-Z，a-z，0-9，‘-’，‘_’，UNICODE字符（\u4E00-\u9FFF）。

响应参数

状态码： 400

表 5-87 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-88 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-89 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-90 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-91 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-92 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-93 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-94 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-95 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-96 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-97 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-98 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-99 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-100 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-101 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-102 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-103 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-104 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-105 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-106 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-107 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-108 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-109 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-110 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-111 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-112 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

删除预定义标签

POST https://{Endpoint}/v1.0/predefine_tags/action

```
{
  "action": "delete",
  "tags": [ {
    "key": "ENV1",
    "value": "DEV1"
  }, {
    "key": "ENV2",
    "value": "DEV2"
  } ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

删除预定义标签

```
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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

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

public class DeletePredefineTagsSolution {

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

TmsClient client = TmsClient.newBuilder()
    .withCredential(auth)
    .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
    .build();
DeletePredefineTagsRequest request = new DeletePredefineTagsRequest();
ReqDeletePredefineTag body = new ReqDeletePredefineTag();
List<PredefineTagRequest> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new PredefineTagRequest()
        .withKey("ENV1")
        .withValue("DEV1")
);
listbodyTags.add(
    new PredefineTagRequest()
        .withKey("ENV2")
        .withValue("DEV2")
);
body.withTags(listbodyTags);
body.withAction(ReqDeletePredefineTag.ActionEnum.fromValue("delete"));
request.withBody(body);
try {
    DeletePredefineTagsResponse response = client.deletePredefineTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

Python

删除预定义标签

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

```

```

try:
    request = DeletePredefineTagsRequest()
    listTagsbody = [
        PredefineTagRequest(
            key="ENV1",
            value="DEV1"
        ),
        PredefineTagRequest(
            key="ENV2",
            value="DEV2"
        )
    ]
    request.body = ReqDeletePredefineTag(
        tags=listTagsbody,
        action="delete"
    )
    response = client.delete_predefine_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

删除预定义标签

```

package main

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

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

    request := &model.DeletePredefineTagsRequest{}
    var listTagsbody = []model.PredefineTagRequest{
        {
            Key: "ENV1",
            Value: "DEV1",
        },
        {
            Key: "ENV2",
            Value: "DEV2",
        },
    }
}

```

```

request.Body = &model.ReqDeletePredefineTag{
    Tags: listTagsbody,
    Action: model.GetReqDeletePredefineTagActionEnum().DELETE,
}
response, err := client.DeletePredefineTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

更多

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

状态码

状态码	描述
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.2.3 查询预定义标签列表

功能介绍

用于查询预定义标签列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/predefine_tags

表 5-113 Query 参数

参数	是否必选	参数类型	描述
key	否	String	键，支持模糊查询，不区分大小写，如果包含“non-URL-safe”的字符，需要进行“urlencoded”。
value	否	String	值，支持模糊查询，不区分大小写，如果包含“non-URL-safe”的字符，需要进行“urlencoded”。
limit	否	Integer	查询记录数。最小为1，最大为1000，未输入时默认为10，为0时不限制查询数据条数。 最小值： 1 最大值： 1000 缺省值： 10
marker	否	String	分页位置标识（索引）。从marker指定索引的下一条数据开始查询。说明：查询第一页数据时，不需要传入此参数，查询后续页码数据时，将查询前一页数据响应体中marker值配入此参数，当返回的tags为空列表时表示查询到最后一页。

参数	是否必选	参数类型	描述
order_field	否	String	排序字段：可输入的值包含（区分大小写）：update_time（更新时间）、key（键）、value（值）。只能选择以上排序字段中的一个，并按照排序方法字段order_method进行排序，如果不传则默认值为：update_time。如以下：若该字段为update_time，则剩余两个默认字段排序为key升序，value升序。若该字段如为key，则剩余两个默认字段排序为update_time降序，value升序。若该字段如为value，则剩余两个默认字段排序为update_time降序，key升序。若该字段不传，默认字段为update_time，则剩余两个默认字段排序为key升序，value升序。 缺省值： update_time
order_method	否	String	order_field字段的排序方法。可输入的值包含（区分大小写）：asc（升序）desc（降序）只能选择以上值的其中之一。不传则默认值为：desc 缺省值： desc

请求参数

表 5-114 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

响应参数

状态码： 200

表 5-115 响应 Body 参数

参数	参数类型	描述
marker	String	分页位置标识（索引）。
total_count	Integer	查询到的标签总数
tags	Array of PredefineTag objects	查询到的标签列表

表 5-116 PredefineTag

参数	参数类型	描述
key	String	键。最大长度36个字符。字符集：A-Z, a-z, 0-9, '-', '_', UNICODE字符（\u4E00-\u9FFF）。
value	String	值。每个值最大长度43个字符，可以为空字符串。字符集：A-Z, a-z, 0-9, '.', '-', '_', UNICODE字符（\u4E00-\u9FFF）。
update_time	String	更新时间，采用UTC时间表示。 2016-12-09T00:00:00Z

状态码： 400

表 5-117 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-118 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-119 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-120 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-121 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-122 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-123 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-124 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-125 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-126 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-127 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-128 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-129 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-130 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-131 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-132 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-133 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-134 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-135 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-136 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-137 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-138 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-139 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-140 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-141 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-142 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

查询预定义标签列表

```
GET https://{Endpoint}/v1.0/predefine_tags?
key=ENV&value=DEV&limit=10&marker=9&order_field=key&order_method=asc
```

响应示例

状态码： 200

OK

```
{
  "marker": "12",
  "total_count": 13,
  "tags": [ {
    "key": "ENV1",
```



```

    "value": "DEV1",
    "update_time": "2017-04-12T14:22:34Z"
  }, {
    "key": "ENV2",
    "value": "DEV2",
    "update_time": "2017-04-12T14:22:34Z"
  }
]
}

```

SDK 代码示例

SDK代码示例如下。

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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

public class ListPredefineTagsSolution {

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

        TmsClient client = TmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListPredefineTagsRequest request = new ListPredefineTagsRequest();
        request.withKey("<key>");
        request.withValue("<value>");
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        request.withOrderField("<order_field>");

        request.withOrderMethod(ListPredefineTagsRequest.OrderMethodEnum.fromValue("<order_method>"));
        try {
            ListPredefineTagsResponse response = client.listPredefineTags(request);
            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 huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

    try:
        request = ListPredefineTagsRequest()
        request.key = "<key>"
        request.value = "<value>"
        request.limit = <limit>
        request.marker = "<marker>"
        request.order_field = "<order_field>"
        request.order_method = "<order_method>"
        response = client.list_predefine_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"
    tms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1/region"
)

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

request := &model.ListPredefineTagsRequest{}
keyRequest:= "<key>"
request.Key = &keyRequest
valueRequest:= "<value>"
request.Value = &valueRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
orderFieldRequest:= "<order_field>"
request.OrderField = &orderFieldRequest
orderMethodRequest:= model.GetListPredefineTagsRequestOrderMethodEnum().<ORDER_METHOD>
request.OrderMethod = &orderMethodRequest
response, err := client.ListPredefineTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

更多

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

状态码

状态码	描述
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented

状态码	描述
503	Service Unavailable

错误码

请参见[错误码](#)。

5.2.4 修改预定义标签

功能介绍

修改预定义标签。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1.0/predefine_tags

请求参数

表 5-143 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

表 5-144 请求 Body 参数

参数	是否必选	参数类型	描述
new_tag	是	PredefineTag Request object	修改后的标签
old_tag	是	PredefineTag Request object	修改前的标签

表 5-145 PredefineTagRequest

参数	是否必选	参数类型	描述
key	是	String	键。最大长度36个字符。字符集：A-Z, a-z, 0-9, '-', '_', UNICODE字符 (\u4E00-\u9FFF)。
value	是	String	值。每个值最大长度43个字符，可以为空字符串。字符集：A-Z, a-z, 0-9, '-', '_', UNICODE字符 (\u4E00-\u9FFF)。

响应参数

状态码： 400

表 5-146 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-147 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-148 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-149 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-150 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-151 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-152 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-153 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-154 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-155 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-156 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-157 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-158 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-159 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-160 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-161 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-162 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-163 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-164 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-165 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-166 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-167 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-168 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-169 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-170 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-171 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

修改预定义标签

PUT https://{Endpoint}/v1.0/predefined_tags

```
{
  "new_tag": {
    "key": "ENV1",
    "value": "DEV1"
  },
  "old_tag": {
    "key": "ENV2",
    "value": "DEV2"
  }
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

修改预定义标签

```
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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

public class UpdatePredefineTagsSolution {

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

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

        UpdatePredefineTagsRequest request = new UpdatePredefineTagsRequest();
        ModifyPredefineTag body = new ModifyPredefineTag();
        PredefineTagRequest oldTagbody = new PredefineTagRequest();
        oldTagbody.withKey("ENV2")
            .withValue("DEV2");
        PredefineTagRequest newTagbody = new PredefineTagRequest();
        newTagbody.withKey("ENV1")
            .withValue("DEV1");
        body.withOldTag(oldTagbody);
        body.withNewTag(newTagbody);
        request.withBody(body);
        try {
            UpdatePredefineTagsResponse response = client.updatePredefineTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

修改预定义标签

```
# coding: utf-8
```

```

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

    try:
        request = UpdatePredefineTagsRequest()
        oldTagbody = PredefineTagRequest(
            key="ENV2",
            value="DEV2"
        )
        newTagbody = PredefineTagRequest(
            key="ENV1",
            value="DEV1"
        )
        request.body = ModifyPrefineTag(
            old_tag=oldTagbody,
            new_tag=newTagbody
        )
        response = client.update_predefine_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

修改预定义标签

```

package main

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

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

request := &model.UpdatePredefineTagsRequest{}
oldTagbody := &model.PredefineTagRequest{
    Key: "ENV2",
    Value: "DEV2",
}
newTagbody := &model.PredefineTagRequest{
    Key: "ENV1",
    Value: "DEV1",
}
request.Body = &model.ModifyPrefineTag{
    OldTag: oldTagbody,
    NewTag: newTagbody,
}
response, err := client.UpdatePredefineTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

更多

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

状态码

状态码	描述
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented

状态码	描述
503	Service Unavailable

错误码

请参见[错误码](#)。

5.3 配额

5.3.1 查询标签配额

功能介绍

查询标签的配额信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/tms/quotas

请求参数

表 5-172 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

响应参数

状态码： 200

表 5-173 响应 Body 参数

参数	参数类型	描述
quotas	Array of TagQuota objects	配额列表

表 5-174 TagQuota

参数	参数类型	描述
quota_key	String	配额key
quota_limit	Integer	配额值
used	Integer	已使用
unit	String	单位

状态码： 400

表 5-175 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-176 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-177 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-178 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-179 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-180 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-181 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-182 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-183 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-184 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-185 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-186 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-187 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-188 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-189 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-190 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-191 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-192 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-193 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-194 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-195 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-196 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-197 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-198 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-199 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-200 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

查询标签配额

GET https://{Endpoint}/v1.0/tms/quotas

响应示例

状态码： 200

OK

```
{
  "quotas": [ {
    "used": 4,
    "unit": "count",
    "quota_key": "predefine_tag",
    "quota_limit": 5000
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

public class ShowTagQuotaSolution {

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

        TmsClient client = TmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTagQuotaRequest request = new ShowTagQuotaRequest();
        try {
            ShowTagQuotaResponse response = client.showTagQuota(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            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 huaweicloudsdkcore.tms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcore.tms.v1 import *

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

    credentials = GlobalCredentials(ak, sk) \

    client = TmsClient.new_builder() \
        .with_credentials(credentials) \

```

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

try:
    request = ShowTagQuotaRequest()
    response = client.show_tag_quota(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.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"
    tms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1/region"
)

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

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

更多

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

状态码

状态码	描述
200	OK

状态码	描述
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.4 资源标签

5.4.1 批量添加标签

功能介绍

用于给云服务的多个资源添加标签，每个资源最多可添加10个标签，每次最多支持批量操作50个资源。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/resource-tags/batch-create

请求参数

表 5-201 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

表 5-202 请求 Body 参数

参数	是否必选	参数类型	描述
project_id	否	String	项目ID，resource_type为region级别服务时为必选项。
resources	是	Array of ResourceTagBody objects	资源列表
tags	是	Array of CreateTagRequest objects	标签列表

表 5-203 ResourceTagBody

参数	是否必选	参数类型	描述
resource_id	是	String	资源ID
resource_type	是	String	资源类型

表 5-204 CreateTagRequest

参数	是否必选	参数类型	描述
key	是	String	键。最大长度36个字符。字符集：A-Z，a-z，0-9，‘-’，‘_’，UNICODE字符（\u4E00-\u9FFF）。

参数	是否必选	参数类型	描述
value	是	String	值。每个值最大长度43个字符，可以为空字符串。字符集：A-Z，a-z，0-9，‘.’，‘-’，‘_’，UNICODE字符（\u4E00-\u9FFF）。

响应参数

状态码： 200

表 5-205 响应 Body 参数

参数	参数类型	描述
failed_resources	Array of TagCreateResponseItem objects	添加标签失败的资源信息

表 5-206 TagCreateResponseItem

参数	参数类型	描述
resource_id	String	资源ID
resource_type	String	资源类型
error_code	String	错误码
error_msg	String	错误描述

状态码： 400

表 5-207 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-208 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-209 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-210 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-211 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-212 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-213 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-214 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-215 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-216 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-217 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-218 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-219 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-220 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-221 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-222 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-223 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-224 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-225 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-226 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-227 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-228 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-229 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-230 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-231 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-232 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

给资源批量添加标签。

```
POST https://{Endpoint}/v1.0/resource-tags/batch-create
```

```
{
  "project_id": "xxxdcffffff",
  "resources": [ {
    "resource_id": "a28531fa-a8d5-468e-8417-86a80962ee5e",
    "resource_type": "disk"
  }, {
    "resource_id": "a28531fa-a8d5-468e-8417-86a8096ddddd",
    "resource_type": "vpc"
  } ],
  "tags": [ {
    "key": "ENV",
    "value": "dev"
  }, {
    "key": "DEPT",
    "value": "pdd"
  } ]
}
```

响应示例

状态码： 200

Success

```
{
  "failed_resources": [ {
    "resource_id": "e1eb7c40cbea4c8389cde527594a306d",
    "resource_type": "disk",
    "error_code": "TMS.0002",
    "error_msg": "Bad request"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

给资源批量添加标签。

```
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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

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

public class CreateResourceTagSolution {

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

TmsClient client = TmsClient.newBuilder()
    .withCredential(auth)
    .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
    .build();
CreateResourceTagRequest request = new CreateResourceTagRequest();
ReqCreateTag body = new ReqCreateTag();
List<CreateTagRequest> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new CreateTagRequest()
        .withKey("ENV")
        .withValue("dev")
);
listbodyTags.add(
    new CreateTagRequest()
        .withKey("DEPT")
        .withValue("pdd")
);
List<ResourceTagBody> listbodyResources = new ArrayList<>();
listbodyResources.add(
    new ResourceTagBody()
        .withResourceId("a28531fa-a8d5-468e-8417-86a80962ee5e")
        .withResourceType("disk")
);
listbodyResources.add(
    new ResourceTagBody()
        .withResourceId("a28531fa-a8d5-468e-8417-86a8096dddd")
        .withResourceType("vpc")
);
body.withTags(listbodyTags);
body.withResources(listbodyResources);
body.withProjectId("xxxxcffffff");
request.withBody(body);
try {
    CreateResourceTagResponse response = client.createResourceTag(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

Python

给资源批量添加标签。

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsktms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsktms.v1 import *

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

try:
    request = CreateResourceTagRequest()
    listTagsbody = [
        CreateTagRequest(
            key="ENV",
            value="dev"
        ),
        CreateTagRequest(
            key="DEPT",
            value="pdd"
        )
    ]
    listResourcesbody = [
        ResourceTagBody(
            resource_id="a28531fa-a8d5-468e-8417-86a80962ee5e",
            resource_type="disk"
        ),
        ResourceTagBody(
            resource_id="a28531fa-a8d5-468e-8417-86a8096dxxxx",
            resource_type="vpc"
        )
    ]
    request.body = ReqCreateTag(
        tags=listTagsbody,
        resources=listResourcesbody,
        project_id="xxxxcffffff"
    )
    response = client.create_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

给资源批量添加标签。

```

package main

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

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

request := &model.CreateResourceTagRequest{}
var listTagsbody = []model.CreateTagRequest{
    {
        Key: "ENV",
        Value: "dev",
    },
    {
        Key: "DEPT",
        Value: "pdd",
    },
}
var listResourcesbody = []model.ResourceTagBody{
    {
        ResourceId: "a28531fa-a8d5-468e-8417-86a80962ee5e",
        ResourceType: "disk",
    },
    {
        ResourceId: "a28531fa-a8d5-468e-8417-86a8096ddddd",
        ResourceType: "vpc",
    },
}
projectIdReqCreateTag:= "xxxxcffffff"
request.Body = &model.ReqCreateTag{
    Tags: listTagsbody,
    Resources: listResourcesbody,
    ProjectId: &projectIdReqCreateTag,
}
response, err := client.CreateResourceTag(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

更多

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

状态码

状态码	描述
200	Success
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found

状态码	描述
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.4.2 批量移除标签

功能介绍

用于批量移除云服务多个资源的标签，每个资源最多支持移除10个标签，每次最多支持批量操作50个资源。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/resource-tags/batch-delete

请求参数

表 5-233 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

表 5-234 请求 Body 参数

参数	是否必选	参数类型	描述
project_id	否	String	项目ID, resource_type为region级别服务时为必选项。
resources	是	Array of ResourceTagBody objects	资源列表
tags	是	Array of DeleteTagRequest objects	标签列表

表 5-235 ResourceTagBody

参数	是否必选	参数类型	描述
resource_id	是	String	资源ID
resource_type	是	String	资源类型

表 5-236 DeleteTagRequest

参数	是否必选	参数类型	描述
key	是	String	键。最大长度36个字符。字符集: A-Z, a-z, 0-9, '-', '_', UNICODE字符 (\u4E00-\u9FFF)。
value	否	String	值。每个值最大长度43个字符, 可以为空字符串。字符集: A-Z, a-z, 0-9, '-', '_', UNICODE字符 (\u4E00-\u9FFF)。

响应参数

状态码: 200

表 5-237 响应 Body 参数

参数	参数类型	描述
failed_resources	Array of TagDeleteResponseItem objects	删除标签失败的资源信息

表 5-238 TagDeleteResponseItem

参数	参数类型	描述
resource_id	String	资源ID
resource_type	String	资源类型
error_code	String	错误码
error_msg	String	错误描述

状态码： 400

表 5-239 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-240 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-241 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-242 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-243 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-244 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-245 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-246 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-247 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-248 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-249 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-250 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-251 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-252 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-253 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-254 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-255 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-256 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-257 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-258 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-259 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-260 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-261 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-262 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-263 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-264 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

批量删除标签。

```
POST https://{Endpoint}/v1.0/resource-tags/batch-delete

{
  "project_id": "xxxdcffffff",
  "resources": [ {
    "resource_id": "a28531fa-a8d5-468e-8417-86a80962ee5e",
    "resource_type": "disk"
  }, {
    "resource_id": "vpc-dc7d19b7",
    "resource_type": "vpc"
  } ],
  "tags": [ {
    "key": "ENV"
  } ]
}
```

响应示例

状态码： 200

Success

```
{
  "failed_resources": [ {
    "resource_id": "e1eb7c40cbea4c8389cde527594a306d",
    "resource_type": "disk",
```

```
"error_code" : "TMS.0002",
"error_msg" : "Bad request"
}]
}
```

SDK 代码示例

SDK代码示例如下。

Java

批量删除标签。

```
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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

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

public class DeleteResourceTagSolution {

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

        TmsClient client = TmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteResourceTagRequest request = new DeleteResourceTagRequest();
        ReqDeleteTag body = new ReqDeleteTag();
        List<DeleteTagRequest> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new DeleteTagRequest()
                .withKey("ENV")
        );
        List<ResourceTagBody> listbodyResources = new ArrayList<>();
        listbodyResources.add(
            new ResourceTagBody()
                .withResourceId("a28531fa-a8d5-468e-8417-86a80962ee5e")
                .withResourceType("disk")
        );
        listbodyResources.add(
            new ResourceTagBody()
                .withResourceId("vpc-dc7d19b7")
                .withResourceType("vpc")
        );
        body.withTags(listbodyTags);
        body.withResources(listbodyResources);
        body.withProjectId("xxxdcffffff");
```

```

request.withBody(body);
try {
    DeleteResourceTagResponse response = client.deleteResourceTag(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

Python

批量删除标签。

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

    try:
        request = DeleteResourceTagRequest()
        listTagsbody = [
            DeleteTagRequest(
                key="ENV"
            )
        ]
        listResourcesbody = [
            ResourceTagBody(
                resource_id="a28531fa-a8d5-468e-8417-86a80962ee5e",
                resource_type="disk"
            ),
            ResourceTagBody(
                resource_id="vpc-dc7d19b7",
                resource_type="vpc"
            )
        ]
        request.body = ReqDeleteTag(
            tags=listTagsbody,
            resources=listResourcesbody,
            project_id="xxxxcffffff"
        )
        response = client.delete_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

批量删除标签。

```
package main

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

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

    request := &model.DeleteResourceTagRequest{}
    var listTagsbody = []model.DeleteTagRequest{
        {
            Key: "ENV",
        },
    }
    var listResourcesbody = []model.ResourceTagBody{
        {
            ResourceId: "a28531fa-a8d5-468e-8417-86a80962ee5e",
            ResourceType: "disk",
        },
        {
            ResourceId: "vpc-dc7d19b7",
            ResourceType: "vpc",
        },
    }
    projectIdReqDeleteTag:= "xxxdcffffff"
    request.Body = &model.ReqDeleteTag{
        Tags: listTagsbody,
        Resources: listResourcesbody,
        ProjectId: &projectIdReqDeleteTag,
    }
    response, err := client.DeleteResourceTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	Success
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.4.3 查询标签键列表

功能介绍

查询指定区域的所有标签键。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/tag-keys

表 5-265 Query 参数

参数	是否必选	参数类型	描述
region_id	否	String	区域ID
limit	否	Integer	查询记录数。最小为1，最大为200，未输入时默认为200。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页位置标识（索引）。从marker指定索引的下一条数据开始查询。说明：查询第一页数据时，不需要传入此参数，查询后续页码数据时，将查询前一页数据响应体中marker值配入此参数，当返回的next_marker为空时表示查询到最后一页。

请求参数

表 5-266 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

响应参数

状态码：200

表 5-267 响应 Body 参数

参数	参数类型	描述
keys	Array of strings	标签键列表
page_info	PageInfoTag Keys object	分页信息

表 5-268 PageInfoTagKeys

参数	参数类型	描述
next_marker	String	分页位置标识（索引）
current_count	Integer	当前页标签键的数量

状态码： 400

表 5-269 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-270 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-271 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-272 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-273 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-274 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-275 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-276 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-277 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-278 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-279 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-280 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-281 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-282 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-283 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-284 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-285 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-286 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-287 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-288 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-289 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-290 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-291 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-292 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-293 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-294 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

查询标签键列表。

```
GET https://{Endpoint}/v1.0/tag-keys?limit=10&marker=xxx
```

响应示例

状态码： 200

OK

```
{
  "keys": [ "ENV1", "ENV2" ],
  "page_info": [ {
    "next_marker": "xxxxx",
    "current_count": 10
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

public class ListTagKeysSolution {

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

```

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

TmsClient client = TmsClient.newBuilder()
    .withCredential(auth)
    .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
    .build();
ListTagKeysRequest request = new ListTagKeysRequest();
request.withRegionId("<region_id>");
request.withLimit(<limit>);
request.withMarker("<marker>");
try {
    ListTagKeysResponse response = client.listTagKeys(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    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 huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

    try:
        request = ListTagKeysRequest()
        request.region_id = "<region_id>"
        request.limit = <limit>
        request.marker = "<marker>"
        response = client.list_tag_keys(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)

```

```
print(e.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"
    tms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/tms/v1/region"
)

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

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

更多

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

状态码

状态码	描述
200	OK
400	Bad Request
401	Unauthorized

状态码	描述
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.4.4 查询标签值列表

功能介绍

查询指定区域的标签键下的所有标签值。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/tag-values

表 5-295 Query 参数

参数	是否必选	参数类型	描述
region_id	否	String	区域ID，存在时查询该区域标签值列表

参数	是否必选	参数类型	描述
limit	否	Integer	查询记录数。最小为1，最大为200，未输入时默认为200。 最小值：1 最大值：200 缺省值：200
marker	否	String	分页位置标识（索引）。从marker指定索引的下一条数据开始查询。说明：查询第一页数据时，不需要传入此参数，查询后续页码数据时，将查询前一页数据响应体中marker值配入此参数，当返回的next_marker为空时表示查询到最后一页。
key	是	String	标签键

请求参数

表 5-296 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

响应参数

状态码： 200

表 5-297 响应 Body 参数

参数	参数类型	描述
values	Array of strings	查询到的标签值列表
page_info	PageInfoTag Values object	分页信息

表 5-298 PageInfoTagValues

参数	参数类型	描述
next_marker	String	分页位置标识（索引）
current_count	Integer	当前页标签值的数量

状态码： 400

表 5-299 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-300 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-301 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-302 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-303 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-304 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-305 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-306 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-307 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-308 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-309 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-310 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-311 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-312 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-313 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-314 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-315 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-316 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-317 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-318 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-319 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-320 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-321 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-322 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-323 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-324 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

查询标签值列表。

```
GET https://{Endpoint}/v1.0/tag-values?limit=10&marker=9
```

响应示例

状态码： 200

OK

```
{
  "values": [ "ENV1", "ENV2" ],
  "page_info": [ {
    "next_marker": "xxxxx",
    "current_count": 10
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

public class ListTagValuesSolution {

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

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

TmsClient client = TmsClient.newBuilder()
    .withCredential(auth)
    .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
    .build();
ListTagValuesRequest request = new ListTagValuesRequest();
request.withRegionId("<region_id>");
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withKey("<key>");
try {
    ListTagValuesResponse response = client.listTagValues(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    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 huaweicloudsktms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsktms.v1 import *

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

    try:
        request = ListTagValuesRequest()
        request.region_id = "<region_id>"
        request.limit = <limit>
        request.marker = "<marker>"
        request.key = "<key>"
        response = client.list_tag_values(request)
        print(response)
```



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

Go

```
package main

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

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

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

更多

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

状态码

状态码	描述
200	OK

状态码	描述
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.4.5 查询资源标签

功能介绍

查询单个资源上的标签。

调用方法

请参见[如何调用API](#)。

URI

GET /v2.0/resources/{resource_id}/tags

表 5-325 路径参数

参数	是否必选	参数类型	描述
resource_id	是	String	资源ID

表 5-326 Query 参数

参数	是否必选	参数类型	描述
project_id	否	String	项目ID, region级资源必选。
resource_type	是	String	资源类型

请求参数

表 5-327 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token

响应参数

状态码： 200

表 5-328 响应 Body 参数

参数	参数类型	描述
tags	Array of TagVo objects	资源标签列表

表 5-329 TagVo

参数	参数类型	描述
key	String	键。最大长度36个字符。字符集：A-Z, a-z, 0-9, '-', '_' , UNICODE字符 (\u4E00-\u9FFF)。
value	String	值。最大长度43个字符，可以为空字符串。字符集：A-Z, a-z, 0-9, '.', '-', '_' , UNICODE字符 (\u4E00-\u9FFF)。

状态码： 401

表 5-330 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-331 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-332 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-333 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-334 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-335 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-336 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-337 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-338 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-339 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-340 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-341 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 410

表 5-342 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-343 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 412

表 5-344 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-345 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 429

表 5-346 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-347 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-348 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-349 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-350 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-351 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-352 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-353 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

查询某个资源的标签。

GET https://{Endpoint}/v2.0/resources/xxxx/tags?project_id=xxxx&resource_type=disk

响应示例

状态码： 200

OK

```
{
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
```



```
"value" : "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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

public class ShowResourceTagSolution {

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

        TmsClient client = TmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowResourceTagRequest request = new ShowResourceTagRequest();
        request.withProjectId("<project_id>");
        request.withResourceType("<resource_type>");
        try {
            ShowResourceTagResponse response = client.showResourceTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            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 huaweicloudsdtms.v1.region.tms_region import TmsRegion
```

```

from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsktms.v1 import *

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

    try:
        request = ShowResourceTagRequest()
        request.project_id = "<project_id>"
        request.resource_type = "<resource_type>"
        response = client.show_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

```

package main

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

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

    request := &model.ShowResourceTagRequest{}
    projectIdRequest := "<project_id>"
    request.ProjectId = &projectIdRequest
    request.ResourceType = "<resource_type>"
    response, err := client.ShowResourceTag(request)
    if err == nil {

```

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

更多

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

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
412	Precondition Failed
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

错误码

请参见[错误码](#)。

5.4.6 根据标签过滤资源

功能介绍

根据标签过滤资源。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/resource-instances/filter

请求参数

表 5-354 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

表 5-355 请求 Body 参数

参数	是否必选	参数类型	描述
project_id	否	String	项目ID，resource_type为region级别服务时为必选项。
resource_types	是	Array of strings	资源类型，此参数为可输入的值（区分大小写）。例如：ecs,scaling_group, images, disk,vpcs,security-groups, shared_bandwidth,eip, cdn等，具体请参见“附录-标签支持的资源类型”章节。
tags	是	Array of Tag objects	标签列表
without_any_tag	否	Boolean	是否仅查询未带标签的资源。该字段为true时查询不带标签的资源。
offset	否	Integer	索引位置，从offset指定的下一条数据开始查询，必须为数字，不能为负数，默认为0。
limit	否	Integer	查询记录数，不传默认为200，limit最多为200，最小值为1。

表 5-356 Tag

参数	是否必选	参数类型	描述
key	是	String	键。最大长度36个字符。字符集：A-Z, a-z, 0-9, '-', '_', UNICODE字符 (\u4E00-\u9FFF)。
values	是	Array of strings	值列表。每个值最大长度43个字符，可以为空字符串。字符集：A-Z, a-z, 0-9, '-', '_', UNICODE字符 (\u4E00-\u9FFF)。

响应参数

状态码： 200

表 5-357 响应 Body 参数

参数	参数类型	描述
resources	Array of Resources objects	资源列表
errors	Array of Errors objects	查询异常时返回的错误信息
total_count	Integer	标签下的资源总数

表 5-358 Resources

参数	参数类型	描述
project_id	String	ProjectID
project_name	String	Project名称
resource_detail	Object	资源详情
resource_id	String	资源ID
resource_name	String	资源名称
resource_type	String	资源类型

参数	参数类型	描述
tags	Array of FilterTagResponse objects	资源标签

表 5-359 FilterTagResponse

参数	参数类型	描述
key	String	键。最大长度36个字符。字符集：A-Z, a-z, 0-9, ‘-’, ‘_’, UNICODE字符（\u4E00-\u9FFF）。
value	String	值。最大长度43个字符，可以为空字符串。字符集：A-Z, a-z, 0-9, ‘.’, ‘-’, ‘_’, UNICODE字符（\u4E00-\u9FFF）。

表 5-360 Errors

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述
project_id	String	ProjectID
resource_type	String	资源类型

状态码： 400

表 5-361 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-362 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-363 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-364 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 402

表 5-365 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-366 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-367 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-368 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-369 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-370 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-371 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-372 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-373 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-374 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 407

表 5-375 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-376 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 408

表 5-377 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-378 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-379 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-380 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-381 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-382 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-383 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-384 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 502

表 5-385 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-386 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-387 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-388 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 504

表 5-389 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-390 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

根据标签过滤资源。

```
POST https://v1.0/resource-instances/filter
{
  "project_id": "e1eb7c40cbea4c8389cde527594a306d",
  "resource_types": [ "disk", "ecs" ],
  "tags": [ {
    "key": "env",
    "values": [ "dev", "prod" ]
  }, {
    "key": "test",
    "values": [ "test" ]
  } ],
  "offset": 0,
  "limit": 10
}
```

响应示例

状态码： 200

OK

```
{
  "resources": [ {
    "project_id": "e1eb7c40cbea4c8389cde527594a306d",
    "project_name": "XXXX",
```

```

"resource_type": "disk",
"resource_id": "b621f5ae-b5c1-49d7-a660-752c445434b4",
"resource_name": "lhj1-volume-0001",
"tags": [{
  "key": "ENV",
  "value": "dev"
}]
}, {
  "project_id": "e1eb7c40cbea4c8389cde527594a306d",
  "project_name": "XXXX",
  "resource_type": "disk",
  "resource_id": "87c9edc9-f66c-48b8-a22f-372b2e22d579",
  "resource_name": "lhj2-volume-0002",
  "tags": [{
    "key": "prod",
    "value": "disk"
  }]
}],
"errors": [{
  "project_id": "e1eb7c40cbea4c8389cde527594a306d",
  "resource_type": "disk",
  "error_code": "TMS.0002",
  "error_msg": "Bad request"
}],
"total_count": 2
}

```

SDK 代码示例

SDK代码示例如下。

Java

根据标签过滤资源。

```

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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

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

public class ListResourceSolution {

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

        TmsClient client = TmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(TmsRegion.valueOf("<YOUR REGION>"))

```

```

        .build();
        ListResourceRequest request = new ListResourceRequest();
        ResqTagResource body = new ResqTagResource();
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("test");
        List<String> listTagsValues1 = new ArrayList<>();
        listTagsValues1.add("dev");
        listTagsValues1.add("prod");
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("env")
                .withValues(listTagsValues1)
        );
        listbodyTags.add(
            new Tag()
                .withKey("test")
                .withValues(listTagsValues)
        );
        List<String> listbodyResourceTypes = new ArrayList<>();
        listbodyResourceTypes.add("disk");
        listbodyResourceTypes.add("ecs");
        body.withLimit(10);
        body.withOffset(0);
        body.withTags(listbodyTags);
        body.withResourceTypes(listbodyResourceTypes);
        body.withProjectId("e1eb7c40cbea4c8389cde527594a306d");
        request.withBody(body);
        try {
            ListResourceResponse response = client.listResource(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}

```

Python

根据标签过滤资源。

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

```

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

try:
    request = ListResourceRequest()
    listValuesTags = [
        "test"
    ]
    listValuesTags1 = [
        "dev",
        "prod"
    ]
    listTagsbody = [
        Tag(
            key="env",
            values=listValuesTags1
        ),
        Tag(
            key="test",
            values=listValuesTags
        )
    ]
    listResourceTypesbody = [
        "disk",
        "ecs"
    ]
    request.body = ResqTagResource(
        limit=10,
        offset=0,
        tags=listTagsbody,
        resource_types=listResourceTypesbody,
        project_id="e1eb7c40cbea4c8389cde527594a306d"
    )
    response = client.list_resource(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

根据标签过滤资源。

```
package main

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

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

request := &model.ListResourceRequest{}
var listValuesTags = []string{
    "test",
}
var listValuesTags1 = []string{
    "dev",
    "prod",
}
var listTagsbody = []model.Tag{
    {
        Key: "env",
        Values: listValuesTags1,
    },
    {
        Key: "test",
        Values: listValuesTags,
    },
}
var listResourceTypesbody = []string{
    "disk",
    "ecs",
}
limitResqTagResource:= int32(10)
offsetResqTagResource:= int32(0)
projectIdResqTagResource:= "e1eb7c40cbea4c8389cde527594a306d"
request.Body = &model.ResqTagResource{
    Limit: &limitResqTagResource,
    Offset: &offsetResqTagResource,
    Tags: listTagsbody,
    ResourceTypes: listResourceTypesbody,
    ProjectId: &projectIdResqTagResource,
}
response, err := client.ListResource(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

更多

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

状态码

状态码	描述
200	OK
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden

状态码	描述
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

错误码

请参见[错误码](#)。

5.5 查询标签管理支持的服务

5.5.1 查询标签管理支持的服务

功能介绍

查询标签管理支持的服务。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/tms/providers

表 5-391 Query 参数

参数	是否必选	参数类型	描述
locale	否	String	指定显示语言 缺省值: zh-cn

参数	是否必选	参数类型	描述
limit	否	Integer	查询记录数默认为10, limit最多为200, 最小值为1。
offset	否	Integer	索引位置, 从offset指定的下一条数据开始查询, 必须为数字, 不能为负数, 默认为0。
provider	否	String	云服务名称

请求参数

表 5-392 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。说明：由于TMS是全局服务，因此调用IAM服务获取用户Token接口时需要注意作用域即scope字段应该设置为domain。响应消息头中X-Subject-Token的值即为用户Token。

响应参数

状态码： 200

表 5-393 响应 Body 参数

参数	参数类型	描述
providers	Array of ProviderResponseBody objects	云服务列表
total_count	Integer	当前支持的云服务总数

表 5-394 ProviderResponseBody

参数	参数类型	描述
provider	String	云服务名称

参数	参数类型	描述
provider_i18n_display_name	String	服务显示名称，可以通过参数中“locale”设置语言。
resource_types	Array of ResourceTypeBody objects	资源类型列表

表 5-395 ResourceTypeBody

参数	参数类型	描述
resource_type	String	资源类型
resource_type_i18n_display_name	String	资源类型显示名称，可以通过参数中“locale”设置语言。
regions	Array of strings	支持的region列表
global	Boolean	是否是全局类型的资源

状态码： 400

表 5-396 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-397 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 401

表 5-398 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-399 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 402

表 5-400 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-401 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 403

表 5-402 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-403 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 404

表 5-404 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-405 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 405

表 5-406 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-407 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 406

表 5-408 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-409 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 407

表 5-410 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-411 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 408

表 5-412 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-413 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 409

表 5-414 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-415 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 500

表 5-416 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-417 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 501

表 5-418 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-419 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 502

表 5-420 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-421 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 503

表 5-422 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-423 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

状态码： 504

表 5-424 响应 Body 参数

参数	参数类型	描述
error	RespErrorMessage object	响应错误信息

表 5-425 RespErrorMessage

参数	参数类型	描述
error_code	String	请求错误码
error_msg	String	错误信息

请求示例

无

响应示例

状态码： 200

OK

```
{
  "providers": [ {
    "provider": "evs",
    "provider_i18n_display_name": "Elastic Volume Service",
    "resource_types": {
      "resource_type_i18n_display_name": "EVS-Disk",
      "global": false,
      "resource_type": "disk",
      "regions": [ "regionId1" ]
    }
  }
],
  "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.tms.v1.region.TmsRegion;
import com.huaweicloud.sdk.tms.v1.*;
import com.huaweicloud.sdk.tms.v1.model.*;

public class ListProvidersSolution {

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

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

        TmsClient client = TmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(TmsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListProvidersRequest request = new ListProvidersRequest();
        request.withLocale(ListProvidersRequest.LocaleEnum.fromValue("<locale>"));
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withProvider("<provider>");
        try {
            ListProvidersResponse response = client.listProviders(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}

```

Python

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdtms.v1.region.tms_region import TmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdtms.v1 import *

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

try:
    request = ListProvidersRequest()
    request.locale = "<locale>"
    request.limit = <limit>
    request.offset = <offset>
    request.provider = "<provider>"
    response = client.list_providers(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

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

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

    request := &model.ListProvidersRequest{}
    localeRequest := model.GetListProvidersRequestLocaleEnum().<LOCALE>
    request.Locale = &localeRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    providerRequest := "<provider>"
    request.Provider = &providerRequest
    response, err := client.ListProviders(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```

```

    } else {
        fmt.Println(err)
    }
}

```

更多

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

状态码

状态码	描述
200	OK
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

错误码

请参见[错误码](#)。

6 权限策略与授权项

6.1 策略及授权项说明

如果您需要对您所拥有的标签管理服务（Tag Management Service）进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用标签管理服务的其它功能。

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略，才能使用户组中的用户获得策略定义的权限，这一过程称为授权。授权后，用户就可以基于策略对云服务进行操作。

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

📖 说明

如果您要允许或是禁止某个接口的操作权限，请使用策略。

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

支持的授权项

细粒度策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。
- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。

- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。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。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](#)。

6.2 标签管理服务 API 授权列表项

表 6-1 TMS 授权项列表

权限	对应API接口	授权项(Action)	IAM项目 (Project)	企业项目 (Enterprise Project)
查询预定义标签	GET /v1.0/predefine_tags	tms:predefineTags:list	支持	不支持
创建预定义标签	POST /v1.0/predefine_tags/action	tms:predefineTags:create	支持	不支持
删除预定义标签	POST /v1.0/predefine_tags/action	tms:predefineTags:delete	支持	不支持
修改预定义标签	PUT /v1.0/predefine_tags	tms:predefineTags:update	支持	不支持
批量创建标签	POST /v1.0/resource-tags/batch-create	tms:resourceTags:create	支持	不支持
批量删除标签	POST /v1.0/resource-tags/batch-delete	tms:resourceTags:delete	支持	不支持
查询标签键列表	GET /v1.0/tag-keys	tms:tagKeys:list	支持	不支持
查询标签值列表	GET /v1.0/tag-values	tms:tagValues:list	支持	不支持
查询资源标签	GET /v2.0/resources/{resource_id}/tags	tms:resourceTags:list	支持	不支持
根据标签过滤资源	POST /v1.0/resource-instances/filter	tms:resources:list	支持	不支持
查询标签配额	GET /v1.0/tms/quotas	无细粒度授权项，但依赖Tenant Guest权限。	不支持	不支持

权限	对应API接口	授权项(Action)	IAM项目 (Project)	企业项目 (Enterprise Project)
查询标签管理支持的服务	GET /v1.0/tms/providers	无细粒度授权项，但依赖Tenant Guest权限。	不支持	不支持

A 附录

A.1 标签支持的资源类型

TMS支持的资源类型（resource_types）可通过如下方式获取：

通过调用[查询标签管理支持的服务](#)接口查询TMS支持的云服务、资源和区域列表。其中provider字段为云服务名称，resource_types中的resource_type字段为资源类型名称。

A.2 状态码

- 正常

返回值	说明
200 OK	GET和PUT操作正常返回。
201 Created	POST操作正常返回。
202 Accepted	请求已被接受。
204 No Content	正常返回。

- 异常

返回值	说明
400 Bad Request	服务器未能处理请求。
401 Unauthorized	被请求的页面需要用户名和密码。
403 Forbidden	禁止访问被请求页面。
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。

返回值	说明
407 Proxy Authentication Required	用户必须首先使用代理服务器进行验证，这样请求才会被处理。
408 Request Timeout	请求超出了服务器的等待时间。
409 Conflict	由于冲突，请求无法被完成。
500 Internal Server Error	请求未完成。服务异常。
501 Not Implemented	请求未完成。服务器不支持所请求的功能。
502 Bad Gateway	请求未完成。服务器从上游服务器收到一个无效的响应。
503 Service Unavailable	请求未完成。系统暂时异常。
504 Gateway Timeout	网关超时。

A.3 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	TMS.0002	Bad request.	请求数据非法	请检查请求参数
400	TMS.0007	Limit is invalid.	limit不合法	请检查limit参数
400	TMS.0008	Marker is invalid.	marker不合法	请检查marker参数
400	TMS.0009	Key is invalid.	key不合法	请检查key参数
400	TMS.0010	Value is invalid.	value不合法	请检查value参数
400	TMS.0011	Action is invalid.	action不合法	请检查action参数
400	TMS.0012	Tags is empty.	tags为空	请检查tags参数
400	TMS.0013	Empty element in tags.	tags含有空元素	请检查tags参数
400	TMS.0016	Values is too much.	value超过最大限额	请检查value参数
400	TMS.0017	Offset is invalid.	offset不合法	请检查offset参数

状态码	错误码	错误信息	描述	处理措施
400	TMS.1001	The number of predefine tag exceeds the upper limit.	预定义标签达到限额	请删除无用的预定义标签
400	TMS.1002	Old_tag cannot be found.	旧标签不存在	请检查旧标签参数
400	TMS.1003	New_tag already exists.	新标签已存在	请检查新标签参数
400	TMS.1004	Old_tag is empty.	旧标签为空	请检查旧标签参数
400	TMS.1005	Invalid key in old_tag.	旧标签中key无效	请检查旧标签Key参数
400	TMS.1006	Invalid value in old_tag.	旧标签中value无效	请检查旧标签Value参数
400	TMS.1007	New_tag is empty.	新标签为空	请检查新标签参数
400	TMS.1008	Invalid key in new_tag.	新标签中key无效	请检查新标签Key参数
400	TMS.1009	Invalid value in new_tag.	新标签中value无效	请检查新标签Value参数
400	TMS.1010	Order_field is invalid.	sortField不合法	请检查sortField参数
400	TMS.1011	Order_method is invalid.	orderMethod不合法	请检查orderMethod参数
401	TMS.0003	Unauthorized user.	用户未被授权	请检查认证token
403	TMS.0004	Permission error.	权限不足	请检查权限
403	TMS.0006	The request is too much, try again later.	请求过多，稍后重试	请求过多，稍后重试
404	TMS.0005	Requested resources not found.	资源未找到	请联系服务支持人员确认接口是否注册
409	TMS.0014	Conflict	内部冲突	请联系服务支持人员

状态码	错误码	错误信息	描述	处理措施
500	TMS.0001	System error.	系统错误	请联系服务支持人员
504	TMS.0018	Query Time Out.	查询超时	请稍后再试

A.4 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 A-1 查看项目 ID



B 修订记录

版本日期	变更说明
2023-07-10	第四次正式发布。 新增 标签支持的资源类型 章节。
2022-12-09	第三次正式发布。 <ul style="list-style-type: none">新增资源标签相关接口。新增查询标签管理支持的服务接口。
2021-11-30	第二次正式发布。 新增 查询标签配额 接口。
2019-02-15	第一次正式发布。