

云审计服务

# API 参考

文档版本 1.0  
发布日期 2024-11-08



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

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

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

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

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

# 目录

<b>1 使用前必读</b>	<b>1</b>
<b>2 API 概览</b>	<b>5</b>
<b>3 如何调用 API</b>	<b>6</b>
3.1 构造请求	6
3.2 认证鉴权	10
3.3 返回结果	11
<b>4 快速入门</b>	<b>13</b>
<b>5 应用示例</b>	<b>15</b>
5.1 示例 1: 创建管理类追踪器	15
5.2 示例 2: 查询追踪器列表	16
5.3 示例 3: 查询管理类事件	17
<b>6 API V3 (推荐使用)</b>	<b>19</b>
6.1 事件管理	19
6.1.1 查询事件列表	19
6.2 追踪器管理	31
6.2.1 创建追踪器	31
6.2.2 修改追踪器	46
6.2.3 查询追踪器	59
6.2.4 删除追踪器	67
6.3 关键操作通知管理	72
6.3.1 创建关键操作通知	72
6.3.2 修改关键操作通知	89
6.3.3 删除关键操作通知	102
6.3.4 查询关键操作通知	107
6.4 其它接口	115
6.4.1 查询租户追踪器配额信息	115
6.4.2 查询云服务的全量操作列表	120
6.4.3 查询审计日志的操作用户列表	126
6.4.4 检查已经配置 OBS 桶是否可以成功转储	131
6.4.5 查询事件的资源类型列表	138
6.5 标签管理	143

6.5.1 批量添加 CTS 资源标签.....	143
6.5.2 批量删除 CTS 资源标签.....	149
<b>7 历史 API.....</b>	<b>155</b>
7.1 API 版本号管理.....	155
7.1.1 查询 API 所有版本号.....	155
7.1.2 查询 API 指定版本号.....	157
7.2 事件管理.....	159
7.2.1 查询事件列表(v1.0).....	160
7.2.2 查询事件列表(v2.0).....	165
7.3 追踪器管理.....	171
7.3.1 创建追踪器.....	171
7.3.2 查询追踪器.....	175
7.3.3 删除追踪器.....	178
7.3.4 修改追踪器.....	179
<b>8 权限和授权项.....</b>	<b>183</b>
<b>9 附录.....</b>	<b>189</b>
9.1 错误码.....	189
9.2 获取账号 ID 和项目 ID.....	193
<b>A 修订记录.....</b>	<b>195</b>

# 1 使用前必读

欢迎使用云审计服务（Cloud Trace Service，以下简称CTS），CTS是云安全解决方案中专业的日志审计服务，提供对各种云资源操作记录的收集、存储和查询功能，可用于支撑安全分析、合规审计、资源跟踪和问题定位等常见应用场景。追踪器创建成功后

您可以使用本文档提供API对云审计服务进行相关操作，如创建、删除追踪器等。支持的全部操作请参见[API概览](#)。

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

## 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](#)中查询云审计服务的终端节点。

云审计服务的终端节点如下表所示，请您根据业务需要选择对应区域的终端节点。

表 1-1 云审计服务的终端节点

区域名称	区域	终端节点（Endpoint）
非洲-约翰内斯堡	af-south-1	cts.af-south-1.myhuaweicloud.com
中国-香港	ap-southeast-1	cts.ap-southeast-1.myhuaweicloud.com
亚太-曼谷	ap-southeast-2	cts.ap-southeast-2.myhuaweicloud.com
亚太-新加坡	ap-southeast-3	cts.ap-southeast-3.myhuaweicloud.com
亚太-雅加达	ap-southeast-4	cts.ap-southeast-4.myhuaweicloud.com
华东-上海二	cn-east-2	cts.cn-east-2.myhuaweicloud.com

区域名称	区域	终端节点 (Endpoint)
华东-上海一	cn-east-3	cts.cn-east-3.myhuaweicloud.com
华北-北京一	cn-north-1	cts.cn-north-1.myhuaweicloud.com
华北-北京二	cn-north-2	cts.cn-north-2.myhuaweicloud.com
华北-北京四	cn-north-4	cts.cn-north-4.myhuaweicloud.com
华北-乌兰察布一	cn-north-9	cts.cn-north-9.myhuaweicloud.com
华南-广州	cn-south-1	cts.cn-south-1.myhuaweicloud.com
华南-深圳	cn-south-2	cts.cn-south-2.myhuaweicloud.com
华南-广州-友好用户环境	cn-south-4	cts.cn-south-4.myhuaweicloud.com
西南-贵阳一	cn-southwest-2	cts.cn-southwest-2.myhuaweicloud.com
欧洲-巴黎	eu-west-0	cts.eu-west-0.myhuaweicloud.com
拉美-墨西哥城二	la-north-2	cts.la-north-2.myhuaweicloud.com
拉美-圣地亚哥	la-south-2	cts.la-south-2.myhuaweicloud.com
中东-利雅得	me-east-1	cts.me-east-1.myhuaweicloud.com
亚太-吉隆坡-OP6	my-kualalumpur-1	cts.my-kualalumpur-1.myhuaweicloud.com
拉美-墨西哥城一	na-mexico-1	cts.na-mexico-1.myhuaweicloud.com
俄罗斯-莫斯科二	ru-northwest-2	cts.ru-northwest-2.myhuaweicloud.com
拉美-圣保罗一	sa-brazil-1	cts.sa-brazil-1.myhuaweicloud.com
土耳其-伊斯坦布尔	tr-west-1	cts.tr-west-1.myhuaweicloud.com

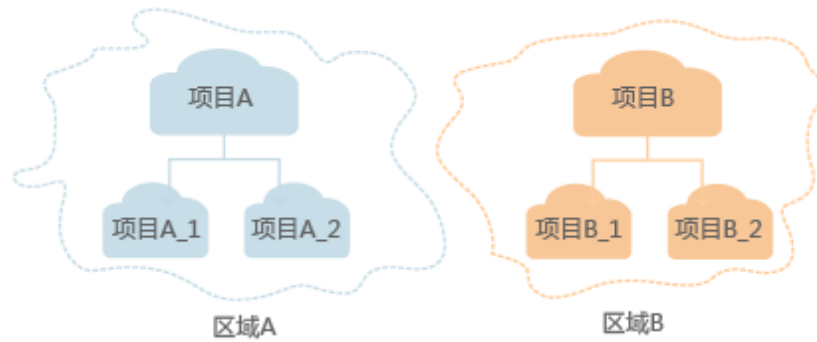
## 约束与限制

- 每个账号可以创建100个数据追踪器和1个管理追踪器，不支持修改配额。
- 更详细的限制请参见具体API的说明。

## 基本概念

- 追踪器  
使用云审计服务前需要开通云审计服务，开通云审计服务时系统会自动创建一个追踪器。该追踪器会自动识别并关联当前租户所使用的所有云服务，并将当前租户的所有操作记录在该追踪器中。  
目前，一个租户仅支持创建1个管理追踪器和100个数据追踪器。
- 事件  
事件即云审计服务追踪并保存的云服务资源的操作日志。您可以通过“事件”了解到谁在什么时间对系统哪些资源做了什么操作。  
事件分为以下两类：管理类事件和数据类事件，管理类事件指云服务上报的事件，数据类事件指OBS服务上报的读写操作事件。
- 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该用户进行日常管理工作。
- 用户  
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。  
在[我的凭证](#)下，您可以查看账号ID和IAM用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。
- 区域（Region）  
从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。  
详情请参见[区域和可用区](#)。
- 可用区（AZ，Availability Zone）  
一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。
- 项目  
区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



同样在[我的凭证](#)下，您可以查看项目ID。

- 企业项目

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

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

## API 版本选择建议

随着云审计服务版本升级，建议您使用功能更强大、操作更便捷的V3版本API。

V1版本API不推荐使用，即将下线。



# 2 API 概览

云审计服务所提供的接口为扩展接口。通过使用云审计服务所提供的接口，您可以完整的使用云审计服务的所有功能。例如查询API版本号、事件列表、创建追踪器等。

云审计服务提供的具体API如[表2-1](#)所示。

**表 2-1** 接口说明

子类型	说明
API版本号	CTS API的版本查询接口，支持查询所有API或者指定API的版本号。
追踪器	CTS API的追踪器管理接口，用来创建、修改、查询以及删除追踪器。
事件	CTS API的事件管理接口，用来查询系统记录的7天内资源操作记录。
关键操作通知	CTS API的关键操作通知管理接口，用来创建、修改、查询以及删除关键操作通知。
标签	CTS API的标签管理接口，用来添加和删除标签。
其它接口	CTS API的其它接口，包含查询租户追踪器配额信息、查询云服务的全量操作列表、查询审计日志的操作用户列表、检查已经配置OBS桶是否可以成功转储、查询事件的资源类型列表。

# 3 如何调用 API

## 3.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的[管理员创建IAM用户](#)接口说明如何调用API。

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

### 请求 URI

请求URI由如下部分组成。

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

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

- **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“华北-北京四”区域的Endpoint为“iam.cn-north-4.myhuaweicloud.com”。
- **resource-path**：资源路径，也即API访问路径。从具体API的URI模块获取，例如“管理员创建IAM用户”API的resource-path为“/v3.0/OS-USER/users”。
- **query-string**：查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

例如您需要获取IAM在“华北-北京四”区域的Token，则需使用“华北-北京四”区域的Endpoint（iam.cn-north-4.myhuaweicloud.com），并在[管理员创建IAM用户](#)的URI部分找到resource-path（/v3.0/OS-USER/users），拼接起来如下所示。

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

图 3-1 URI 示意图



**说明**

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

## 请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

- **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**管理员创建IAM用户**的URI部分，您可以看到其请求方法为“POST”，则其请求为：

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

## 请求消息头

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

详细的公共请求消息头字段请参见[表3-1](#)。

表 3-1 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

名称	描述	是否必选	示例
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
Authorization	签名认证信息，当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。	否	SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date, Signature=*****
X-Sdk-Date	请求发送的时间，当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。	否	20240416T095341Z
X-Project-ID	project id，项目编号。请参考 <a href="#">获取账号ID和项目ID</a> 章节获取项目编号。	否 如果是专属云场景采用AK/SK认证方式的接口请求或者多project场景采用AK/SK认证的接口请求，则该字段必选。	e9993fc787d94b6c886cb aa340f9c0f4
X-Domain-ID	账号ID，全局服务的接口调用时，需在请求消息头中携带X-Domain-ID。	否	b1dd2183937***** *0bda

名称	描述	是否必选	示例
X-Auth-Token	用户Token。 用户Token也就是调用 <a href="#">获取用户Token</a> 接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头（Headers）中包含的“X-Subject-Token”的值即为Token值。	否 使用Token认证时该字段必选。	注：以下仅为Token示例片段 MIIPAgYJKoZlhvcNAQcCo ...ggg1BBIIINPXsidG9rZ

### 📖 说明

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

AK/SK认证的详细说明请参见[认证鉴权](#)的“AK/SK认证”。

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

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

## 请求消息体

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

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

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

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

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

{
  "user": {
```

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

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

## 3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- 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：通过Token认证调用请求。

### AK/SK 认证

#### 📖 说明

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

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

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

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

#### 须知

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

### Token 认证

#### 📖 说明

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

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

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

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

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

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/usersContent-Type: application/json
X-Auth-Token: ABCDEFJ....
```

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

## 3.3 返回结果

### 状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

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

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

### 响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于“管理员创建IAM用户”接口，返回如**图3-2**所示的消息头。

图 3-2 管理员创建 IAM 用户响应消息头

```
"X-Frame-Options": "SAMEORIGIN",  
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fcea8a",  
"Transfer-Encoding": "chunked",  
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",  
"Server": "api-gateway",  
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",  
"X-Content-Type-Options": "nosniff",  
"Connection": "keep-alive",  
"X-Download-Options": "noopen",  
"X-XSS-Protection": "1; mode=block;",  
"X-IAM-Trace-Id": "token_██████████_null_af2953f2bcc67a42325a69a19e6c32a2",  
"Date": "Tue, 21 May 2024 09:03:40 GMT",  
"Content-Type": "application/json; charset=utf8"
```

## 响应消息体

响应消息体通常以结构化格式返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

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

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

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

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

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



# 4 快速入门

本节通过调用CTS创建追踪器接口来创建一个追踪器。

## 📖 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
- CTS创建追踪器的API

## 操作步骤

1. Token认证，具体操作请参考[认证鉴权](#)。
2. 发送“POST /v1.0/{project\_id}/tracker”。
3. 在Request Header中增加“Content-Type”和“X-Auth-Token”
4. 在Request Body中传入参数如下：

```
POST /v1.0/{project_id}/tracker
{
  "bucket_name": "obs-f1da", //选择转储的OBS桶名称（必填，String）
  "is_support_trace_files_encryption": true, //事件文件转储加密功能开关。（可选，布尔）。
  "kms_id": "13a4207c-7abe-4b68-8510-16b84c3b5504", //事件文件转储加密所采用的密钥id。（当
  "is_support_trace_files_encryption"参数值为“true”时，此参数为必选项。）
  "is_obs_created": true, //是否支持新建OBS桶。（可选，布尔）
  "file_prefix_name": "yO8Q", //标示需要存储于OBS的日志文件前缀。（可选，String）
  "log_file_validate": {
    "is_support_validate": true //是否打开事件文件校验。（可选，布尔）
  }
}
```

请求响应成功后，返回已创建追踪器的信息。

```
{
  "bucket_name": "obs-f1da",
  "file_prefix_name": "yO8Q",
  "is_obs_created": true,
  "is_support_trace_files_encryption": true,
  "kms_id": "13a4207c-7abe-4b68-8510-16b84c3b5504",
  "log_file_validate": {
```

```
"is_support_validate": true
},
"tracker_name": "system", //标识追踪器名称
"status": "enabled" //标识追踪器状态。
}
```

若请求失败，则会返回错误码及对应的错误信息说明，详情请参考：[错误码](#)。

# 5 应用示例

## 5.1 示例 1：创建管理类追踪器

### 场景描述

本章以创建一个配置OBS转储桶的管理类追踪器为例。

### 涉及其它云服务接口

创建管理类追踪器时，需要查询用户的桶列表：

**获取桶列表**：确定将要配置转储的桶名称。

### 创建管理类追踪器

- 接口相关信息  
URI格式：POST /v3/{project\_id}/tracker
- 请求示例  
POST https://{cts\_endpoint}/v3/{project\_id}/tracker  
{cts\_endpoint}信息请从[地区和终端节点](#)获取。

Body:

```
{
  "tracker_type": "system",
  "tracker_name": "system",
  "obs_info": {
    "is_obs_created": false,
    "bucket_name": "test_tracker",
    "file_prefix_name": "yO8Q "
  }
}
```

- 响应示例  
{
 "id": "2e6fa9b8-8c6e-456d-b5d3-77be97xxxx",
 "create\_time": 1587958482923,
 "domain\_id": "aexxxxxxxx4d4fb4bexxxxxx791fbf",
 "obs\_info": {
 "bucket\_name": " test\_tracker ",
 "file\_prefix\_name": "yO8Q"
 }
}

```
},  
"project_id": "bb1xxxxxxxxe4f498cbxxxxxxxx35634",  
"tracker_name": "system",  
"tracker_type": "system",  
"status": "enabled"  
}
```

## 5.2 示例 2：查询追踪器列表

### 场景描述

本章为查询一个账号下所有追踪器信息。

### 涉及其它云服务接口

无。

### 查询追踪器

- 接口相关信息  
URI格式：GET /v3/{project\_id}/trackers
- 请求示例  
GET https://{cts\_endpoint}/v3/{project\_id}/trackers  
{cts\_endpoint}信息请从[地区和终端节点](#)获取。
- 响应示例

```
{  
  "trackers": [  
    {  
      "is_support_trace_files_encryption": true,  
      "create_time": 1589886034121,  
      "streamId": "4a1ef2b6-d79a-4dc6-90f0-48151cd5491b",  
      "kms_id": "7dbbb3fa-93e4-4528-bc7b-9beb794b0229",  
      "groupId": "26fa12ac-75f7-42ed-8118-ab9f2263042f",  
      "is_support_validate": false,  
      "obs_info": {  
        "is_obs_created": false,  
        "bucket_name": "",  
        "is_authorized_bucket": false,  
        "file_prefix_name": "",  
        "bucket_lifecycle": 0  
      },  
      "lts": {  
        "log_group_name": "CTS",  
        "is_lts_enabled": true,  
        "log_topic_name": "system-trace"  
      },  
      "tracker_type": "system",  
      "domain_id": "2306579dc99f4c8690b14b68e734fcd9",  
      "project_id": "24edf66e79d04187acb99a463e610764",  
      "tracker_name": "system",  
      "id": "ebf8d1c3-762b-4ce3-b316-6b1aa32f8be3",  
      "status": "enabled"  
    },  
    {  
      "domain_id": "2306579dc99f4c8690b14b68e734fcd9",  
      "is_support_trace_files_encryption": false,  
      "obs_info": {  
        "is_obs_created": false,  
        "bucket_name": "",  
        "is_authorized_bucket": false,  
        "file_prefix_name": "",  
        "bucket_lifecycle": 0  
      },  
      "lts": {  
        "log_group_name": "CTS",  
        "is_lts_enabled": true,  
        "log_topic_name": "system-trace"  
      },  
      "tracker_type": "system",  
      "domain_id": "2306579dc99f4c8690b14b68e734fcd9",  
      "project_id": "24edf66e79d04187acb99a463e610764",  
      "tracker_name": "system",  
      "id": "ebf8d1c3-762b-4ce3-b316-6b1aa32f8be3",  
      "status": "enabled"  
    }  
  ]  
}
```

```
    "is_authorized_bucket": false,
    "file_prefix_name": "",
    "bucket_lifecycle": 0
  },
  "create_time": 1589276171198,
  "project_id": "24edf66e79d04187acb99a463e610764",
  "data_bucket": {
    "data_event": [
      "READ",
      "WRITE"
    ],
    "search_enabled": false,
    "data_bucket_name": "ctest0423"
  },
  "tracker_name": "sdsa",
  "is_support_validate": false,
  "lts": {
    "log_group_name": "CTS",
    "is_lts_enabled": false,
    "log_topic_name": "sdsa"
  },
  "id": "c9a3961d-3aa0-4e60-8e63-dd4ce7f1a88a",
  "status": "enabled",
  "tracker_type": "data"
}
]
```

## 5.3 示例 3：查询管理类事件

### 场景描述

本章以查询租户最近一小时内所有管理类事件为例。

### 涉及其它云服务接口

无。

### 查询管理类事件列表

- 接口相关信息  
URI格式：GET /v3/{project\_id}/traces
- 请求示例  
GET https://{cts\_endpoint}/v3/{project\_id} /traces?trace\_type=system  
{cts\_endpoint}信息请从[地区和终端节点](#)获取。

- 响应示例

```
{
  "meta_data": {
    "count": 2,
    "marker": "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8"
  },
  "traces": [ {
    "time": 1472148708232,
    "user": {
      "name": "xxx",
      "domain": {
        "name": "xxx",
        "id": "ded649d814464428ba89d04d7955c93e"
      }
    }
  },
```

```
"response" : {
  "code" : "VPC.0514",
  "message" : "Update port fail."
},
"code" : 200,
"service_type" : "VPC",
"resource_type" : "eip",
"resource_name" : "192.144.163.1",
"resource_id" : "d502809d-0d1d-41ce-9690-784282142ccc",
"trace_name" : "deleteEip",
"trace_rating" : "warning",
"trace_type" : "ConsoleAction",
"api_version" : "2.0",
"record_time" : 1481066128032,
"trace_id" : "e001ccb9-bc09-11e6-b00b-4b2a61338db6"
}, {
  "time" : 1472148708232,
  "user" : {
    "name" : "xxx",
    "domain" : {
      "name" : "xxx",
      "id" : "ded649d814464428ba89d04d7955c93e"
    }
  }
},
"response" : {
  "code" : "VPC.0514",
  "message" : "Update port fail."
},
"code" : 200,
"service_type" : "VPC",
"resource_type" : "eip",
"resource_name" : "192.144.163.1",
"resource_id" : "d502809d-0d1d-41ce-9690-784282142ccc",
"trace_name" : "deleteEip",
"trace_rating" : "warning",
"trace_type" : "ConsoleAction",
"api_version" : "2.0",
"record_time" : 1481066128032,
"trace_id" : "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8"
}]
}
```

# 6 API V3 (推荐使用)

## 6.1 事件管理

### 6.1.1 查询事件列表

#### 功能介绍

通过事件列表查询接口，可以查出系统记录的7天内资源操作记录。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{project\_id}/traces

表 6-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，参见 <a href="#">获取账号ID和项目ID</a> 章节。

表 6-2 Query 参数

参数	是否必选	参数类型	描述
trace_type	是	String	标识审计事件类型。目前支持管理类事件 (system) 和数据类事件 (data)。默认值为 "system"。 枚举值： <ul style="list-style-type: none"><li>• system</li><li>• data</li></ul>
from	否	Long	标识查询事件列表的起始时间戳 (timestamp, 为标准UTC时间, 毫秒级, 13位数字, 不包括传入时间) 默认为上一小时的时间戳。查询条件from与to配套使用。
next	否	String	取值为响应中marker的值, 用于标识查询事件的起始时间 (自此条事件的记录时间起, 向更早时间查询)。可以与 "from"、"to" 结合使用。最终的查询条件取两组时间条件的交集。
to	否	Long	标识查询事件列表的结束时间戳 (timestamp, 为标准UTC时间, 毫秒级, 13位数字, 不包括传入时间) 默认为当前时间戳。查询条件to与from配套使用。
tracker_name	否	String	当"trace_type"字段值为 "system"时, 该字段值默认为 "system"。当"trace_type"字段值为"data"时, 该字段值可以传入数据类追踪器名称, 达到筛选某个数据类追踪器下的数据事件目的。
service_type	否	String	标识查询事件列表对应的云服务类型。必须为已对接CTS的云服务的英文缩写, 且服务类型一般为大写字母。当"trace_type"字段值为"system"时, 该字段筛选有效"。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节。



参数	是否必选	参数类型	描述
user	否	String	标识特定用户名称, 用以查询该用户下的所有事件。当 "trace_type" 字段值为 "system" 时, 该字段筛选有效。
resource_name	否	String	标示查询事件列表对应的的资源名称。当 "trace_type" 字段值为 "system" 时, 该字段筛选有效。说明: 该字段可能包含大写字母。
resource_type	否	String	标示查询事件列表对应的资源类型。当 "trace_type" 字段值为 "system" 时, 该字段筛选有效。
resource_id	否	String	标示查询事件列表对应的云服务资源ID。当 "trace_type" 字段值为 "system" 时, 该字段筛选有效。
limit	否	Integer	标示查询事件列表中限定返回的事件条数。不传时默认10条, 最大值200条。
trace_id	否	String	标示某一条事件的事件ID。当传入这个查询条件时, 其他查询条件自动不生效。当 "trace_type" 字段值为 "system" 时, 该字段筛选有效。
trace_name	否	String	标示查询事件列表对应的事件名称。当 "trace_type" 字段值为 "system" 时, 该字段筛选有效。说明: 该字段可能包含大写字母。
trace_rating	否	String	标示查询事件列表对应的事件等级目前有三种: 正常(normal), 警告(warning), 事故(incident)。当 "trace_type" 字段值为 "system" 时, 该字段筛选有效。 枚举值: <ul style="list-style-type: none"><li>• normal</li><li>• warning</li><li>• incident</li></ul>
access_key_id	否	String	标示查询事件列表对应的访问密钥ID。包含临时访问凭证和永久访问密钥。

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	标示查询事件列表对应的企业项目ID。

## 请求参数

无

## 响应参数

状态码： 200

表 6-3 响应 Body 参数

参数	参数类型	描述
traces	Array of <a href="#">Traces</a> objects	本次查询事件列表返回事件数组。
meta_data	<a href="#">MetaData</a> object	本次查询事件条数和标记位。

表 6-4 Traces

参数	参数类型	描述
resource_id	String	标识事件对应的云服务资源ID。
trace_name	String	标识查询事件列表对应的事件名称。由0-9,a-z,A-Z, '-', '_', ':' 组成, 长度为1~64个字符, 且以首字符必须为字母。
trace_rating	String	标识事件等级, 目前有三种: 正常 (normal), 警告 (warning), 事故 (incident)。 枚举值: <ul style="list-style-type: none"><li>• normal</li><li>• warning</li><li>• incident</li></ul>
trace_type	String	标识事件发生源头类型, 管理类事件主要包括API调用 (ApiCall), Console页面调用 (ConsoleAction) 和系统间调用 (SystemAction)。数据类事件主要包括ObsSDK, ObsAPI。
request	String	标识事件对应接口请求内容, 即资源操作请求体。

参数	参数类型	描述
response	String	记录用户请求的响应，标识事件对应接口响应内容，即资源操作结果返回体。
code	String	记录用户请求的响应，标识事件对应接口返回的HTTP状态码。
api_version	String	标识事件对应的云服务接口版本。
message	String	标识其他云服务为此条事件添加的备注信息。
record_time	Long	标识云审计服务记录本次事件的时间戳。
trace_id	String	标识事件的ID，由系统生成的UUID。
time	Long	标识事件产生的时间戳。
user	<b>UserInfo</b> object	标识触发事件的用户信息。
service_type	String	标识查询事件列表对应的云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。
resource_type	String	查询事件列表对应的资源类型。
source_ip	String	标识触发事件的租户IP。
resource_name	String	标识事件对应的资源名称。
request_id	String	记录本次请求的request id
location_info	String	记录本次请求出错后，问题定位所需要的辅助信息。
endpoint	String	云资源的详情页面
resource_url	String	云资源的详情页面的访问链接（不含endpoint）
enterprise_project_id	String	标识资源所在的企业项目ID。
resource_account_id	String	标识资源所在的账号ID。仅在跨租户操作资源时有值。
read_only	boolean	标识用户请求是不是只读。
operation_id	String	记录事件对应的操作ID。

表 6-5 UserInfo

参数	参数类型	描述
id	String	用户ID，参见《云审计服务API参考》“获取账号ID和项目ID”章节。

参数	参数类型	描述
name	String	用户名称。
user_name	String	用户名称。
domain	<a href="#">BaseUser</a> object	标识触发事件的用户domain信息。
account_id	String	账号ID, 参见《云审计服务API参考》“获取账号ID和项目ID”章节。
access_key_id	String	访问密钥ID。
principal_urn	String	操作用户身份的 URN。 <ul style="list-style-type: none"><li>如果是 IAM 用户身份, 格式如 iam::<account-id>:user:&lt;user-name&gt;</account-id></li><li>如果是 IAM 委托会话身份, 格式如 sts::sts::<account-id>:assumed-agency:&lt;agency-name&gt;/&lt;agency-session-name&gt;</account-id></li><li>如果是 IAM 联邦身份, 格式如 sts::<account-id>:external-user:&lt;idp_id&gt;/&lt;user-session-name&gt;</account-id></li></ul>
principal_id	String	操作用户身份Id。 <ul style="list-style-type: none"><li>如果是 IAM 用户身份, 格式为 &lt;user-id&gt;</li><li>如果是 IAM 委托会话身份, 格式为 &lt;agency-id&gt;:&lt;agency-session-name&gt;</li><li>如果是 IAM 联邦身份, 格式为 &lt;idp_id&gt;:&lt;user-session-name&gt;</li></ul>
principal_is_root_user	String	是否是根用户。 <ul style="list-style-type: none"><li>值为“true”时, 表示操作者是根用户。</li><li>值为“false”时, 表示操作者是委托会话身份、联邦身份或非根用户的 IAM 用户。</li></ul> 枚举值: <ul style="list-style-type: none"><li><b>true</b></li><li><b>false</b></li></ul>
type	String	操作者身份类型。
invoked_by	Array of strings	发出请求的服务的名称。控制台操作时为 ["service.console"]
session_context	<a href="#">SessionContext</a> object	临时安全凭据属性。

表 6-6 BaseUser

参数	参数类型	描述
id	String	账号ID, 参见《云审计服务API参考》“获取账号ID和项目ID”章节。
name	String	账号名称。

表 6-7 SessionContext

参数	参数类型	描述
attributes	<b>Attributes</b> object	临时安全凭据的属性。

表 6-8 Attributes

参数	参数类型	描述
created_at	String	颁发临时安全凭证时的时间 (timestamp, 为标准UTC时间, 毫秒级, 13位数字)。
mfa_authenticated	String	是否已经通过MFA身份认证。

表 6-9 MetaData

参数	参数类型	描述
count	Integer	标识本次查询事件列表返回的事件记录的总条数。
marker	String	标识本次查询事件列表返回的最后一个事件ID。可以使用这个参数返回值作为分页请求参数next的值, 如果marker返回为null, 则表示当前请求条件下查询事件列表已经全部返回没有下一页。

状态码: 400

表 6-10 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 401**

**表 6-11** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 403**

**表 6-12** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 404**

**表 6-13** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 500**

**表 6-14** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 503**

**表 6-15** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。

参数	参数类型	描述
error_msg	String	错误描述。

## 请求示例

- 查询管理类事件请求样例。  
GET https://{endpoint}/v3/{project\_id}/traces?  
limit=11&to=1479095278000&from=1478490478000&trace\_name=createTracker&resource\_type=tracke  
r&service\_type=CTS&trace\_type=system
- 查询数据类事件请求样例。  
GET https://{endpoint}/v3/{project\_id}/traces?  
limit=11&to=1479095278000&from=1478490478000&trace\_type=data

## 响应示例

状态码：200

请求正常。

```
{
  "meta_data" : {
    "count" : 2,
    "marker" : "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8"
  },
  "traces" : [ {
    "time" : 1472148708232,
    "user" : {
      "name" : "xxx",
      "domain" : {
        "name" : "xxx",
        "id" : "ded649d814464428ba89d04d7955c93e"
      }
    }
  },
  "response" : {
    "code" : "VPC.0514",
    "message" : "Update port fail."
  },
  "code" : 200,
  "service_type" : "VPC",
  "resource_type" : "eip",
  "resource_name" : "192.144.163.1",
  "resource_id" : "d502809d-0d1d-41ce-9690-784282142ccc",
  "trace_name" : "deleteEip",
  "trace_rating" : "warning",
  "trace_type" : "ConsoleAction",
  "api_version" : "2.0",
  "record_time" : 1481066128032,
  "trace_id" : "e001ccb9-bc09-11e6-b00b-4b2a61338db6",
  "read_only": false,
  "operation_id": "ListSubscriptions"
}, {
  "time" : 1472148708232,
  "user" : {
    "name" : "xxx",
    "domain" : {
      "name" : "xxx",
      "id" : "ded649d814464428ba89d04d7955c93e"
    }
  },
  "response" : {
    "code" : "VPC.0514",
    "message" : "Update port fail."
  }
}
```

```
    },
    "code" : 200,
    "service_type" : "VPC",
    "resource_type" : "eip",
    "resource_name" : "192.144.163.1",
    "resource_id" : "d502809d-0d1d-41ce-9690-784282142ccc",
    "trace_name" : "deleteEip",
    "trace_rating" : "warning",
    "trace_type" : "ConsoleAction",
    "api_version" : "2.0",
    "record_time" : 1481066128032,
    "trace_id" : "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8",
    "read_only": false,
    "operation_id": "ListSubscriptions"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

public class ListTracesSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListTracesRequest request = new ListTracesRequest();
        request.withTraceType(ListTracesRequest.TraceTypeEnum.fromValue("<trace_type>"));
        request.withLimit(<limit>);
        request.withFrom(<from>L);
        request.withNext("<next>");
        request.withTo(<to>L);
        request.withTrackerName("<tracker_name>");
        request.withServiceType("<service_type>");
        request.withUser("<user>");
        request.withResourceId("<resource_id>");
        request.withResourceName("<resource_name>");
        request.withResourceType("<resource_type>");
        request.withTraceId("<trace_id>");
        request.withTraceName("<trace_name>");
    }
}
```



```
request.withTraceRating(ListTracesRequest.TraceRatingEnum.fromValue("<trace_rating>"));
request.withAccessKeyld("<access_key_id>");
request.withEnterpriseProjectId("<enterprise_project_id>");
try {
    ListTracesResponse response = client.listTraces(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListTracesRequest()
        request.trace_type = "<trace_type>"
        request.limit = <limit>
        request._from = <from>
        request.next = "<next>"
        request.to = <to>
        request.tracker_name = "<tracker_name>"
        request.service_type = "<service_type>"
        request.user = "<user>"
        request.resource_id = "<resource_id>"
        request.resource_name = "<resource_name>"
        request.resource_type = "<resource_type>"
        request.trace_id = "<trace_id>"
        request.trace_name = "<trace_name>"
        request.trace_rating = "<trace_rating>"
        request.access_key_id = "<access_key_id>"
        request.enterprise_project_id = "<enterprise_project_id>"
        response = client.list_traces(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

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

## Go

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTracesRequest{}
    request.TraceType = model.GetListTracesRequestTraceTypeEnum().<TRACE_TYPE>
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    fromRequest := int64(<from>)
    request.From = &fromRequest
    nextRequest := "<next>"
    request.Next = &nextRequest
    toRequest := int64(<to>)
    request.To = &toRequest
    trackerNameRequest := "<tracker_name>"
    request.TrackerName = &trackerNameRequest
    serviceTypeRequest := "<service_type>"
    request.ServiceType = &serviceTypeRequest
    userRequest := "<user>"
    request.User = &userRequest
    resourceIdRequest := "<resource_id>"
    request.ResourceId = &resourceIdRequest
    resourceNameRequest := "<resource_name>"
    request.ResourceName = &resourceNameRequest
    resourceTypeRequest := "<resource_type>"
    request.ResourceType = &resourceTypeRequest
    traceIdRequest := "<trace_id>"
    request.TraceId = &traceIdRequest
    traceNameRequest := "<trace_name>"
    request.TraceName = &traceNameRequest
    traceRatingRequest := model.GetListTracesRequestTraceRatingEnum().<TRACE_RATING>
    request.TraceRating = &traceRatingRequest
    accessKeyIdRequest := "<access_key_id>"
    request.AccessKeyId = &accessKeyIdRequest
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    response, err := client.ListTraces(request)
    if err == nil {
```

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

## 更多

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

## 状态码

状态码	描述
200	请求正常。
400	查询参数异常，请求未完成。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	查询事件不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.2 追踪器管理

### 6.2.1 创建追踪器

#### 功能介绍

云审计服务开通后系统会自动创建一个追踪器，用来关联系统记录的所有操作。目前，一个云账户在一个Region下支持创建一个管理类追踪器和多个数据类追踪器。云审计服务支持在管理控制台查询近7天内的操作记录。如需保存更长时间的操作记录，您可以在创建追踪器之后通过对象存储服务（Object Storage Service，以下简称OBS）将操作记录实时保存至OBS桶中。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/tracker

表 6-16 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

## 请求参数

表 6-17 请求 Body 参数

参数	是否必选	参数类型	描述
tracker_type	是	String	标识追踪器类型。目前支持系统追踪器类型有管理类追踪器(system)和数据类追踪器(data)。数据类追踪器和管理类追踪器共同参数有: is_lts_enabled, obs_info, is_support_validate; 管理类追踪器参数: is_support_trace_files_encryption, kms_id; 数据类追踪器参数: tracker_name, data_bucket。 枚举值: <ul style="list-style-type: none"><li>• system</li><li>• data</li></ul>
tracker_name	是	String	标识追踪器名称。当"tracker_type"参数值为"system"时该参数为默认值"system"。当"tracker_type"参数值为"data"时该参数需要指定追踪器名称"。
is_lts_enabled	否	Boolean	是否打开事件分析。
is_organization_tracker	否	Boolean	是否应用到我的组织。只针对管理类追踪器。设置为true时, ORG组织下所有成员当前区域的审计日志会转储到该追踪器配置的OBS桶或者LTS日志流, 但是事件列表界面不支持查看其它组织成员的审计日志。
management_event_selector	否	ManagementEventSelector object	管理类事件选择器。

参数	是否必选	参数类型	描述
obs_info	否	<b>TrackerObsInfo</b> object	转储桶配置
is_support_trace_files_encryption	否	Boolean	事件文件转储加密功能开关。当"tracker_type"参数值为"system"时该参数值有效。该参数必须与kms_id参数同时使用。
kms_id	否	String	事件文件转储加密所采用的密钥id (从KMS获取)。当"tracker_type"参数值为"system"时该参数值有效。当"is_support_trace_files_encryption"参数值为“是”时，此参数为必选项。
is_support_validate	否	Boolean	事件文件转储时是否打开事件文件校验。
data_bucket	否	<b>DataBucket</b> object	追踪桶配置信息。当"tracker_type"参数值为"data"时该参数值有效。
agency_name	否	String	云服务委托名称。参数值为"cts_admin_trust"时，创建追踪器会自动创建云服务委托：cts_admin_trust。 枚举值： <ul style="list-style-type: none"> <li>• cts_admin_trust</li> </ul>

表 6-18 ManagementEventSelector

参数	是否必选	参数类型	描述
exclude_service	否	Array of strings	标识不转储的云服务名称。目前只支持设置为KMS，表示屏蔽KMS服务的createDatakey事件。

表 6-19 TrackerObsInfo

参数	是否必选	参数类型	描述
bucket_name	否	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。

参数	是否必选	参数类型	描述
file_prefix_name	否	String	标识需要存储于OBS的日志文件前缀，0-9，a-z，A-Z，'-','!','_'长度为0~64字符。
is_obs_created	否	Boolean	是否支持新建OBS桶。值为“true”时，表示新创建OBS桶存储事件文件；值为“false”时，选择已存在的OBS桶存储事件文件。
bucket_lifecycle	否	Integer	标识配置桶内对象存储周期。当“tracker_type”参数值为“data”时该参数值有效。 枚举值： <ul style="list-style-type: none"><li>• 30</li><li>• 60</li><li>• 90</li><li>• 180</li><li>• 1095</li></ul>
compress_type	否	String	压缩类型。包括不压缩（json），压缩（gzip）两种状态。默认为gzip格式。 枚举值： <ul style="list-style-type: none"><li>• gzip</li><li>• json</li></ul>
is_sort_by_service	否	Boolean	路径按云服务划分，打开后转储文件路径中将增加云服务名。默认为true。

表 6-20 DataBucket

参数	是否必选	参数类型	描述
data_bucket_name	否	String	数据类追踪器追踪对象的桶名。 <ul style="list-style-type: none"><li>• 当启用或者停用数据类追踪器时，该参数为必选。</li><li>• 管理类追踪器无此参数。</li><li>• 追踪器一旦创建追踪桶无法修改。</li></ul>

参数	是否必选	参数类型	描述
data_event	否	Array of strings	<p>数据类追踪器追踪的操作类型。</p> <ul style="list-style-type: none"> <li>当启用或者停用数据类追踪器时，该参数为必选。</li> <li>管理类追踪器无此参数。</li> <li>READ OBS对象读取操作；WRITE OBS对象写操作。</li> </ul> <p>枚举值：</p> <ul style="list-style-type: none"> <li><b>WRITE</b></li> <li><b>READ</b></li> </ul>

## 响应参数

状态码： 201

表 6-21 响应 Body 参数

参数	参数类型	描述
id	String	追踪器唯一标识。
create_time	Long	追踪器创建时间戳。
kms_id	String	事件文件转储加密所采用的密钥id（从KMS获取）。当"tracker_type"参数值为"system"和"is_support_trace_files_encryption"参数值为“是”时，此参数为必选项。
is_support_validation	Boolean	是否打开事件文件校验。
is_organization_tracker	Boolean	是否应用到我的组织。只针对管理类追踪器。设置为true时，ORG组织下所有成员当前区域的审计日志会转储到该追踪器配置的OBS桶或者LTS日志流，但是事件列表界面不支持查看其它组织成员的审计日志。
management_event_selector	<b>ManagementEventSelector</b> object	管理类事件选择器。
lts	<b>Lts</b> object	事件分析
tracker_type	String	<p>标识追踪器类型。目前支持系统追踪器类型有管理类追踪器（system）和数据类追踪器（data）。</p> <p>枚举值：</p> <ul style="list-style-type: none"> <li><b>system</b></li> <li><b>data</b></li> </ul>

参数	参数类型	描述
domain_id	String	账号ID, 参见《云审计服务API参考》“获取账号ID和项目ID”章节。
project_id	String	项目ID。
tracker_name	String	标识追踪器名称, 当前版本默认为“system”。
status	String	标识追踪器状态, 包括正常 (enabled), 停止 (disabled) 和异常 (error) 三种状态, 状态为异常时需通过明细 (detail) 字段说明错误来源。 枚举值: <ul style="list-style-type: none"> <li>enabled</li> <li>disabled</li> </ul>
detail	String	该参数仅在追踪器状态异常时返回, 用于标识追踪器异常的原因, 包括桶策略异常 (bucketPolicyError), 桶不存在 (noBucket) 和欠费或冻结 (arrears) 三种原因。
is_support_trace_files_encryption	Boolean	事件文件转储加密功能开关。该参数必须与 kms_id 参数同时使用。当前环境仅 "tracker_type" 参数值为 "system" 时支持该功能。
obs_info	ObsInfo object	事件转储桶信息。
data_bucket	DataBucketQuery object	数据类事件追踪桶信息。当 "tracker_type" 参数值为 "data" 时有效。
agency_name	String	云服务委托名称。参数值为 "cts_admin_trust" 时, 创建追踪器会自动创建云服务委托: cts_admin_trust。 枚举值: <ul style="list-style-type: none"> <li>cts_admin_trust</li> </ul>

表 6-22 ManagementEventSelector

参数	参数类型	描述
exclude_service	Array of strings	标识不转储的云服务名称。目前只支持设置为 KMS, 表示屏蔽 KMS 服务的 createDatakey 事件。

表 6-23 Lts

参数	参数类型	描述
is_lts_enabled	Boolean	是否启用日志服务检索功能。



参数	参数类型	描述
log_group_name	String	云审计服务在日志服务中创建的日志组名称。
log_topic_name	String	云审计服务在日志服务中创建的日志主题名称。

表 6-24 ObsInfo

参数	参数类型	描述
bucket_name	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。
file_prefix_name	String	标识需要存储于OBS的日志文件前缀，0-9，a-z，A-Z，'!'，'!'，'_'长度为0~64字符。
is_obs_created	Boolean	标识配置桶是否由追踪器自动创建。
is_authorized_bucket	Boolean	标识配置桶是否已经授权给CTS服务账号。
bucket_lifecycle	Long	标识配置桶内对象存储周期。当"tracker_type"参数值为"data"时该参数值有效。
compress_type	String	压缩类型。包括不压缩 (json)，压缩 (gzip) 两种状态。默认为gzip格式。 枚举值： <ul style="list-style-type: none"><li>• gzip</li><li>• json</li></ul>
is_sort_by_service	Boolean	路径按云服务划分，打开后转储文件路径中将增加云服务名。默认为true。

表 6-25 DataBucketQuery

参数	参数类型	描述
data_bucket_name	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。
search_enabled	Boolean	追踪桶日志是否支持搜索。

参数	参数类型	描述
data_event	Array of strings	<p>数据类追踪器追踪对象的桶名。</p> <ul style="list-style-type: none"> <li>当启用或者停用数据类追踪器时，该参数为必选。</li> <li>管理类追踪器无此参数。</li> <li>追踪器一旦创建追踪桶无法修改。</li> <li>READ OBS对象读取操作；WRITE OBS对象写操作。</li> </ul> <p>枚举值：</p> <ul style="list-style-type: none"> <li>WRITE</li> <li>READ</li> </ul>

状态码： 400

表 6-26 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

状态码： 401

表 6-27 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

状态码： 403

表 6-28 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

状态码： 404

表 6-29 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 500

表 6-30 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 503

表 6-31 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

- 管理类追踪器创建样例。  
POST https://{endpoint}/v3/{project\_id}/tracker  

```
{
  "tracker_type": "system",
  "tracker_name": "system",
  "agency_name": "cts_admin_trust",
  "obs_info": {
    "is_obs_created": false,
    "bucket_name": "test-data-tracker",
    "file_prefix_name": "11"
  },
  "is_lts_enabled": true,
  "is_support_trace_files_encryption": true,
  "kms_id": "13a4207c-7abe-4b68-8510-16b84c3b5504",
  "is_support_validate": true
}
```
- 数据类追踪器创建样例。  

```
{
  "tracker_type": "data",
  "tracker_name": "data-tracker-name",
  "agency_name": "cts_admin_trust",
  "obs_info": {
    "is_obs_created": false,
    "bucket_name": "saveTraceBucket",

```

```
"file_prefix_name" : "11",
"bucket_lifecycle" : 30
},
"is_lts_enabled" : true,
"data_bucket" : {
  "data_event" : [ "READ", "WRITE" ],
  "data_bucket_name" : "cstest0423"
}
}
```

## 响应示例

**状态码： 201**

请求成功。

```
{
  "id" : "2e6fa9b8-8c6e-456d-b5d3-77be972d220b",
  "create_time" : 1587958482923,
  "domain_id" : "aexxxxxxxxx4d4fb4bexxxxxxxxx791fbf",
  "is_support_trace_files_encryption" : true,
  "kms_id" : "13a4207c-7abe-4b68-8510-16b84c3b5504",
  "agency_name" : "cts_admin_trust",
  "obs_info" : {
    "is_obs_created" : false,
    "bucket_name" : "test-bucket",
    "is_authorized_bucket" : false,
    "file_prefix_name" : "11",
    "bucket_lifecycle" : 30
  },
  "project_id" : "bb1xxxxxxxxe4f498cbxxxxxxxx35634",
  "lts" : {
    "is_lts_enabled" : true,
    "log_group_name" : "CTS",
    "log_topic_name" : "system-trace"
  },
  "is_support_validate" : true,
  "tracker_name" : "system",
  "tracker_type" : "system",
  "status" : "enabled"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 管理类追踪器创建样例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

public class CreateTrackerSolution {

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

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

CtsClient client = CtsClient.newBuilder()
    .withCredential(auth)
    .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
    .build();
CreateTrackerRequest request = new CreateTrackerRequest();
CreateTrackerRequestBody body = new CreateTrackerRequestBody();
TrackerObsInfo obsInfoBody = new TrackerObsInfo();
obsInfoBody.withBucketName("test-data-tracker")
    .withFilePrefixName("11")
    .withIsObsCreated(false);
body.withIsSupportValidate(true);
body.withKmsId("13a4207c-7abe-4b68-8510-16b84c3b5504");
body.withIsSupportTraceFilesEncryption(true);
body.withObsInfo(obsInfoBody);
body.withIsLtsEnabled(true);

body.withAgencyName(CreateTrackerRequestBody.AgencyNameEnum.fromValue("cts_admin_trust"));
body.withTrackerName("system");
body.withTrackerType(CreateTrackerRequestBody.TrackerTypeEnum.fromValue("system"));
request.withBody(body);
try {
    CreateTrackerResponse response = client.createTracker(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 数据类追踪器创建样例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

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

public class CreateTrackerSolution {

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

running this example, set environment variables CLOUD\_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 BasicCredentials()
    .withAk(ak)
    .withSk(sk);

CtsClient client = CtsClient.newBuilder()
    .withCredential(auth)
    .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
    .build();
CreateTrackerRequest request = new CreateTrackerRequest();
CreateTrackerRequestBody body = new CreateTrackerRequestBody();
List<DataBucket.DataEventEnum> listDataBucketDataEvent = new ArrayList<>();
listDataBucketDataEvent.add(DataBucket.DataEventEnum.fromValue("READ"));
listDataBucketDataEvent.add(DataBucket.DataEventEnum.fromValue("WRITE"));
DataBucket dataBucketbody = new DataBucket();
dataBucketbody.withDataBucketName("cstest0423")
    .withDataEvent(listDataBucketDataEvent);
TrackerObsInfo obsInfobody = new TrackerObsInfo();
obsInfobody.withBucketName("saveTraceBucket")
    .withFilePrefixName("11")
    .withIsObsCreated(false)
    .withBucketLifecycle(30);
body.withDataBucket(dataBucketbody);
body.withObsInfo(obsInfobody);
body.withIsLtsEnabled(true);

body.withAgencyName(CreateTrackerRequestBody.AgencyNameEnum.fromValue("cts_admin_trust"));
body.withTrackerName("data-tracker-name");
body.withTrackerType(CreateTrackerRequestBody.TrackerTypeEnum.fromValue("data"));
request.withBody(body);
try {
    CreateTrackerResponse response = client.createTracker(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

- 管理类追踪器创建样例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]

credentials = BasicCredentials(ak, sk)

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

try:
    request = CreateTrackerRequest()
    obsInfobody = TrackerObsInfo(
        bucket_name="test-data-tracker",
        file_prefix_name="11",
        is_obs_created=False
    )
    request.body = CreateTrackerRequestBody(
        is_support_validate=True,
        kms_id="13a4207c-7abe-4b68-8510-16b84c3b5504",
        is_support_trace_files_encryption=True,
        obs_info=obsInfobody,
        is_lts_enabled=True,
        agency_name="cts_admin_trust",
        tracker_name="system",
        tracker_type="system"
    )
    response = client.create_tracker(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 数据类追踪器创建样例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = CreateTrackerRequest()
        listDataEventDataBucket = [
            "READ",
            "WRITE"
        ]
        dataBucketbody = DataBucket(
            data_bucket_name="ctest0423",
            data_event=listDataEventDataBucket
        )
```

```
)
obsInfobody = TrackerObsInfo(
    bucket_name="saveTraceBucket",
    file_prefix_name="11",
    is_obs_created=False,
    bucket_lifecycle=30
)
request.body = CreateTrackerRequestBody(
    data_bucket=dataBucketbody,
    obs_info=obsInfobody,
    is_lts_enabled=True,
    agency_name="cts_admin_trust",
    tracker_name="data-tracker-name",
    tracker_type="data"
)
response = client.create_tracker(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

- 管理类追踪器创建样例。

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build()
    )

    request := &model.CreateTrackerRequest{
        bucketNameObsInfo:= "test-data-tracker"
        filePrefixNameObsInfo:= "11"
        isObsCreatedObsInfo:= false
        obsInfobody := &model.TrackerObsInfo{
            BucketName: &bucketNameObsInfo,
            FilePrefixName: &filePrefixNameObsInfo,
            IsObsCreated: &isObsCreatedObsInfo,
        }
    }
    isSupportValidateCreateTrackerRequestBody:= true
    kmsIdCreateTrackerRequestBody:= "13a4207c-7abe-4b68-8510-16b84c3b5504"
    isSupportTraceFilesEncryptionCreateTrackerRequestBody:= true
```



```
isLtsEnabledCreateTrackerRequestBody:= true
agencyNameCreateTrackerRequestBody:=
model.GetCreateTrackerRequestBodyAgencyNameEnum().CTS_ADMIN_TRUST
request.Body = &model.CreateTrackerRequestBody{
    IsSupportValidate: &isSupportValidateCreateTrackerRequestBody,
    KmsId: &kmsIdCreateTrackerRequestBody,
    IsSupportTraceFilesEncryption: &isSupportTraceFilesEncryptionCreateTrackerRequestBody,
    ObsInfo: obsInfobody,
    IsLtsEnabled: &isLtsEnabledCreateTrackerRequestBody,
    AgencyName: &agencyNameCreateTrackerRequestBody,
    TrackerName: "system",
    TrackerType: model.GetCreateTrackerRequestBodyTrackerTypeEnum().SYSTEM,
}
response, err := client.CreateTracker(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 数据类追踪器创建样例。

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateTrackerRequest{}
    var listDataEventDataBucket = []model.DataBucketDataEvent{
        model.GetDataBucketDataEventEnum().READ,
        model.GetDataBucketDataEventEnum().WRITE,
    }
    dataBucketNameDataBucket:= "ctest0423"
    dataBucketbody := &model.DataBucket{
        DataBucketName: &dataBucketNameDataBucket,
        DataEvent: &listDataEventDataBucket,
    }
    bucketNameObsInfo:= "saveTraceBucket"
    filePrefixNameObsInfo:= "11"
    isObsCreatedObsInfo:= false
    bucketLifecycleObsInfo:= int32(30)
    obsInfobody := &model.TrackerObsInfo{
        BucketName: &bucketNameObsInfo,
        FilePrefixName: &filePrefixNameObsInfo,
    }
```

```
    IsObsCreated: &isObsCreatedObsInfo,  
    BucketLifecycle: &bucketLifecycleObsInfo,  
  }  
  isLtsEnabledCreateTrackerRequestBody:= true  
  agencyNameCreateTrackerRequestBody:=  
model.GetCreateTrackerRequestBodyAgencyNameEnum().CTS_ADMIN_TRUST  
  request.Body = &model.CreateTrackerRequestBody{  
    DataBucket: dataBucketbody,  
    ObsInfo: obsInfobody,  
    IsLtsEnabled: &isLtsEnabledCreateTrackerRequestBody,  
    AgencyName: &agencyNameCreateTrackerRequestBody,  
    TrackerName: "data-tracker-name",  
    TrackerType: model.GetCreateTrackerRequestBodyTrackerTypeEnum().DATA,  
  }  
  response, err := client.CreateTracker(request)  
  if err == nil {  
    fmt.Printf("%+v\n", response)  
  } else {  
    fmt.Println(err)  
  }  
}
```

## 更多

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

## 状态码

状态码	描述
201	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求中的资源不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.2.2 修改追踪器

### 功能介绍

云审计服务支持修改已创建追踪器的配置项，包括OBS桶转储、关键事件通知、事件转储加密、通过LTS对管理类事件进行检索、事件文件完整性校验以及追踪器启停状态等相关参数，修改追踪器对已有的操作记录没有影响。修改追踪器完成后，系统立即以新的规则开始记录操作。

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v3/{project\_id}/tracker

表 6-32 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

## 请求参数

表 6-33 请求 Body 参数

参数	是否必选	参数类型	描述
tracker_type	是	String	标识追踪器类型。目前支持系统追踪器类型有管理类追踪器(system)和数据类追踪器(data)。数据类追踪器和管理类追踪器共同参数有: is_lts_enabled, obs_info, is_support_validate; 管理类追踪器参数: is_support_trace_files_encryption, kms_id; 数据类追踪器参数: tracker_name, data_bucket。 枚举值: <ul style="list-style-type: none"><li>• system</li><li>• data</li></ul>
tracker_name	是	String	标识追踪器名称。当"tracker_type"参数值为"system"时该参数为默认值"system"。当"tracker_type"参数值为"data"时该参数需要指定追踪器名称"。

参数	是否必选	参数类型	描述
status	否	String	标识追踪器状态，该接口中可修改的状态包括正常（enabled）和停止（disabled）。如果选择修改状态为停止，则修改成功后追踪器停止记录事件。 枚举值： <ul style="list-style-type: none"><li>• enabled</li><li>• disabled</li></ul>
is_lts_enabled	否	Boolean	是否打开事件分析。
is_organization_tracker	否	Boolean	是否应用到我的组织。只针对管理类追踪器。设置为true时，ORG组织下所有成员当前区域的审计日志会转储到该追踪器配置的OBS桶或者LTS日志流，但是事件列表界面不支持查看其它组织成员的审计日志。
management_event_selector	否	<b>ManagementEventSelector</b> object	管理类事件选择器。
obs_info	否	<b>TrackerObsInfo</b> object	转储桶配置
is_support_trace_files_encryption	否	Boolean	事件文件转储加密功能开关。当"tracker_type"参数值为"system"时该参数值有效。该参数必须与kms_id参数同时使用。
kms_id	否	String	事件文件转储加密所采用的密钥id（从KMS获取）。当"tracker_type"参数值为"system"时该参数值有效。当"is_support_trace_files_encryption"参数值为“是”时，此参数为必选项。
is_support_validate	否	Boolean	事件文件转储时是否打开事件文件校验。
data_bucket	否	<b>DataBucket</b> object	追踪桶配置信息。当"tracker_type"参数值为"data"时该参数值有效。

参数	是否必选	参数类型	描述
agency_name	否	String	云服务委托名称。参数值为 "cts_admin_trust" 时，修改追踪器会自动创建云服务委托：cts_admin_trust。 枚举值： <ul style="list-style-type: none"> <li>cts_admin_trust</li> </ul>

表 6-34 ManagementEventSelector

参数	是否必选	参数类型	描述
exclude_service	否	Array of strings	标识不转储的云服务名称。目前只支持设置为KMS，表示屏蔽KMS服务的createDatakey事件。

表 6-35 TrackerObsInfo

参数	是否必选	参数类型	描述
bucket_name	否	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。
file_prefix_name	否	String	标识需要存储于OBS的日志文件前缀，0-9, a-z, A-Z, '-', '!', '_' 长度为0~64字符。
is_obs_created	否	Boolean	是否支持新建OBS桶。值为“true”时，表示新创建OBS桶存储事件文件；值为“false”时，选择已存在的OBS桶存储事件文件。
bucket_lifecycle	否	Integer	标识配置桶内对象存储周期。当"tracker_type"参数值为"data"时该参数值有效。 枚举值： <ul style="list-style-type: none"> <li>30</li> <li>60</li> <li>90</li> <li>180</li> <li>1095</li> </ul>

参数	是否必选	参数类型	描述
compress_type	否	String	压缩类型。包括不压缩（json），压缩（gzip）两种状态。默认为gzip格式。 枚举值： <ul style="list-style-type: none"><li>• gzip</li><li>• json</li></ul>
is_sort_by_service	否	Boolean	路径按云服务划分，打开后转储文件路径中将增加云服务名。默认为true。

表 6-36 DataBucket

参数	是否必选	参数类型	描述
data_bucket_name	否	String	数据类追踪器追踪对象的桶名。 <ul style="list-style-type: none"><li>• 当启用或者停用数据类追踪器时，该参数为必选。</li><li>• 管理类追踪器无此参数。</li><li>• 追踪器一旦创建追踪桶无法修改。</li></ul>
data_event	否	Array of strings	数据类追踪器追踪的操作类型。 <ul style="list-style-type: none"><li>• 当启用或者停用数据类追踪器时，该参数为必选。</li><li>• 管理类追踪器无此参数。</li><li>• READ OBS对象读取操作；WRITE OBS对象写操作。</li></ul> 枚举值： <ul style="list-style-type: none"><li>• WRITE</li><li>• READ</li></ul>

## 响应参数

状态码：400

表 6-37 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

**状态码： 401**

**表 6-38** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 403**

**表 6-39** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 404**

**表 6-40** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 500**

**表 6-41** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 503**

**表 6-42** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。

参数	参数类型	描述
error_msg	String	错误描述。

## 请求示例

- 管理类追踪器修改样例。  
PUT https://{endpoint}/v3/{project\_id}/tracker

```
{
  "tracker_type": "system",
  "tracker_name": "system",
  "agency_name": "cts_admin_trust",
  "obs_info": {
    "is_obs_created": false,
    "bucket_name": "test-data-tracker",
    "file_prefix_name": "11"
  },
  "is_lts_enabled": false,
  "is_support_trace_files_encryption": false,
  "kms_id": "",
  "is_support_validate": false,
  "status": "enabled"
}
```

- 数据类追踪器修改样例。

```
{
  "tracker_type": "data",
  "tracker_name": "data-tracker-name",
  "agency_name": "cts_admin_trust",
  "obs_info": {
    "is_obs_created": false,
    "bucket_name": "",
    "file_prefix_name": "",
    "bucket_lifecycle": 60
  },
  "is_lts_enabled": true,
  "data_bucket": {
    "data_event": [ "READ", "WRITE" ]
  }
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

- 管理类追踪器修改样例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
```



```
import com.huaweicloud.sdk.cts.v3.model.*;

public class UpdateTrackerSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTrackerRequest request = new UpdateTrackerRequest();
        UpdateTrackerRequestBody body = new UpdateTrackerRequestBody();
        TrackerObsInfo obsInfobody = new TrackerObsInfo();
        obsInfobody.withBucketName("test-data-tracker")
            .withFilePrefixName("11")
            .withIsObsCreated(false);
        body.withIsSupportValidate(false);
        body.withKmsId("");
        body.withIsSupportTraceFilesEncryption(false);
        body.withObsInfo(obsInfobody);
        body.withIsLtsEnabled(false);
        body.withStatus(UpdateTrackerRequestBody.StatusEnum.fromValue("enabled"));

        body.withAgencyName(UpdateTrackerRequestBody.AgencyNameEnum.fromValue("cts_admin_trust"));
        body.withTrackerName("system");
        body.withTrackerType(UpdateTrackerRequestBody.TrackerTypeEnum.fromValue("system"));
        request.withBody(body);
        try {
            UpdateTrackerResponse response = client.updateTracker(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 数据类追踪器修改样例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

import java.util.List;
```

```
import java.util.ArrayList;

public class UpdateTrackerSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTrackerRequest request = new UpdateTrackerRequest();
        UpdateTrackerRequestBody body = new UpdateTrackerRequestBody();
        List<DataBucket.DataEventEnum> listDataBucketDataEvent = new ArrayList<>();
        listDataBucketDataEvent.add(DataBucket.DataEventEnum.fromValue("READ"));
        listDataBucketDataEvent.add(DataBucket.DataEventEnum.fromValue("WRITE"));
        DataBucket dataBucketbody = new DataBucket();
        dataBucketbody.withDataEvent(listDataBucketDataEvent);
        TrackerObsInfo obsInfobody = new TrackerObsInfo();
        obsInfobody.withBucketName("")
            .withFilePrefixName("")
            .withIsObsCreated(false)
            .withBucketLifecycle(60);
        body.withDataBucket(dataBucketbody);
        body.withObsInfo(obsInfobody);
        body.withIsLtsEnabled(true);

        body.withAgencyName(UpdateTrackerRequestBody.AgencyNameEnum.fromValue("cts_admin_trust"));
        body.withTrackerName("data-tracker-name");
        body.withTrackerType(UpdateTrackerRequestBody.TrackerTypeEnum.fromValue("data"));
        request.withBody(body);
        try {
            UpdateTrackerResponse response = client.updateTracker(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

- 管理类追踪器修改样例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *
```

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = UpdateTrackerRequest()
        obsInfobody = TrackerObsInfo(
            bucket_name="test-data-tracker",
            file_prefix_name="11",
            is_obs_created=False
        )
        request.body = UpdateTrackerRequestBody(
            is_support_validate=False,
            kms_id="",
            is_support_trace_files_encryption=False,
            obs_info=obsInfobody,
            is_lts_enabled=False,
            status="enabled",
            agency_name="cts_admin_trust",
            tracker_name="system",
            tracker_type="system"
        )
        response = client.update_tracker(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 数据类追踪器修改样例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

```
try:
    request = UpdateTrackerRequest()
    listDataEventDataBucket = [
        "READ",
        "WRITE"
    ]
    dataBucketbody = DataBucket(
        data_event=listDataEventDataBucket
    )
    obsInfobody = TrackerObsInfo(
        bucket_name="",
        file_prefix_name="",
        is_obs_created=False,
        bucket_lifecycle=60
    )
    request.body = UpdateTrackerRequestBody(
        data_bucket=dataBucketbody,
        obs_info=obsInfobody,
        is_lts_enabled=True,
        agency_name="cts_admin_trust",
        tracker_name="data-tracker-name",
        tracker_type="data"
    )
    response = client.update_tracker(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

- 管理类追踪器修改样例。

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTrackerRequest{}
    bucketNameObsInfo := "test-data-tracker"
    filePrefixNameObsInfo := "11"
    isObsCreatedObsInfo := false
```

```
obsInfobody := &model.TrackerObsInfo{
    BucketName: &bucketNameObsInfo,
    FilePrefixName: &filePrefixNameObsInfo,
    IsObsCreated: &isObsCreatedObsInfo,
}
isSupportValidateUpdateTrackerRequestBody:= false
kmsIdUpdateTrackerRequestBody:= ""
isSupportTraceFilesEncryptionUpdateTrackerRequestBody:= false
isLtsEnabledUpdateTrackerRequestBody:= false
statusUpdateTrackerRequestBody:= model.GetUpdateTrackerRequestBodyStatusEnum().ENABLED
agencyNameUpdateTrackerRequestBody:=
model.GetUpdateTrackerRequestBodyAgencyNameEnum().CTS_ADMIN_TRUST
request.Body = &model.UpdateTrackerRequestBody{
    IsSupportValidate: &isSupportValidateUpdateTrackerRequestBody,
    KmsId: &kmsIdUpdateTrackerRequestBody,
    IsSupportTraceFilesEncryption: &isSupportTraceFilesEncryptionUpdateTrackerRequestBody,
    ObsInfo: obsInfobody,
    IsLtsEnabled: &isLtsEnabledUpdateTrackerRequestBody,
    Status: &statusUpdateTrackerRequestBody,
    AgencyName: &agencyNameUpdateTrackerRequestBody,
    TrackerName: "system",
    TrackerType: model.GetUpdateTrackerRequestBodyTrackerTypeEnum().SYSTEM,
}
response, err := client.UpdateTracker(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 数据类追踪器修改样例。

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTrackerRequest{}
    var listDataEventDataBucket = []model.DataBucketDataEvent{
        model.GetDataBucketDataEventEnum().READ,
        model.GetDataBucketDataEventEnum().WRITE,
    }
    dataBucketbody := &model.DataBucket{
        DataEvent: &listDataEventDataBucket,
    }
```

```
}
bucketNameObsInfo:= ""
filePrefixNameObsInfo:= ""
isObsCreatedObsInfo:= false
bucketLifecycleObsInfo:= int32(60)
obsInfobody := &model.TrackerObsInfo{
    BucketName: &bucketNameObsInfo,
    FilePrefixName: &filePrefixNameObsInfo,
    IsObsCreated: &isObsCreatedObsInfo,
    BucketLifecycle: &bucketLifecycleObsInfo,
}
isLtsEnabledUpdateTrackerRequestBody:= true
agencyNameUpdateTrackerRequestBody:=
model.GetUpdateTrackerRequestBodyAgencyNameEnum().CTS_ADMIN_TRUST
request.Body = &model.UpdateTrackerRequestBody{
    DataBucket: dataBucketbody,
    ObsInfo: obsInfobody,
    IsLtsEnabled: &isLtsEnabledUpdateTrackerRequestBody,
    AgencyName: &agencyNameUpdateTrackerRequestBody,
    TrackerName: "data-tracker-name",
    TrackerType: model.GetUpdateTrackerRequestBodyTrackerTypeEnum().DATA,
}
response, err := client.UpdateTracker(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	请求正常。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	服务器无法找到被请求的资源。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.2.3 查询追踪器

### 功能介绍

开通云审计服务成功后，您可以在追踪器信息页面查看追踪器的详细信息。详细信息主要包括追踪器名称，用于存储操作事件的OBS桶名称和OBS桶中的事件文件前缀。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/trackers

表 6-43 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，参见 <a href="#">获取账号ID和项目ID</a> 章节。

表 6-44 Query 参数

参数	是否必选	参数类型	描述
tracker_name	否	String	标示追踪器名称。在不传入该字段的情况下，将查询租户所有的追踪器。
tracker_type	否	String	标识追踪器类型。目前支持系统追踪器有管理类追踪器（system）和数据类追踪器（data）。 枚举值： <ul style="list-style-type: none"><li>• system</li><li>• data</li></ul>

### 请求参数

无

### 响应参数

状态码： 200

表 6-45 响应 Body 参数

参数	参数类型	描述
trackers	Array of <b>TrackerResponseBody</b> objects	本次查询追踪器列表返回的追踪器数组。

表 6-46 TrackerResponseBody

参数	参数类型	描述
id	String	追踪器唯一标识。
create_time	Long	追踪器创建时间戳。
kms_id	String	事件文件转储加密所采用的密钥id (从KMS获取)。当"tracker_type"参数值为"system"和"is_support_trace_files_encryption"参数值为“是”时,此参数为必选项。
is_support_validate	Boolean	是否打开事件文件校验。
lts	<b>Lts</b> object	事件分析
tracker_type	String	标识追踪器类型。目前支持系统追踪器类型有管理类追踪器 (system) 和数据类追踪器 (data)。 枚举值: <ul style="list-style-type: none"><li>• <b>system</b></li><li>• <b>data</b></li></ul>
domain_id	String	账号ID, 参见《云审计服务API参考》“获取账号ID和项目ID”章节。
project_id	String	项目ID。
tracker_name	String	标识追踪器名称, 当前版本默认为“system”。
status	String	标识追踪器状态, 包括正常 (enabled), 停止 (disabled) 和异常 (error) 三种状态, 状态为异常时需通过明细 (detail) 字段说明错误来源。 枚举值: <ul style="list-style-type: none"><li>• <b>enabled</b></li><li>• <b>disabled</b></li></ul>
detail	String	该参数仅在追踪器状态异常时返回, 用于标识追踪器异常的原因, 包括桶策略异常 (bucketPolicyError), 桶不存在 (noBucket) 和欠费或冻结 (arrears) 三种原因。



参数	参数类型	描述
is_support_trace_files_encryption	Boolean	事件文件转储加密功能开关。该参数必须与 kms_id 参数同时使用。当前环境仅 "tracker_type" 参数值为 "system" 时支持该功能。
obs_info	<b>ObsInfo</b> object	事件转储桶信息。
data_bucket	<b>DataBucketQuery</b> object	数据类事件追踪桶信息。当 "tracker_type" 参数值为 "data" 时有效。
group_id	String	LTS 服务日志组的 ID。
stream_id	String	LTS 服务日志流的 ID。
is_organization_tracker	Boolean	是否应用到我的组织。只针对管理类追踪器。设置为 true 时，ORG 组织下所有成员当前区域的审计日志会转储到该追踪器配置的 OBS 桶或者 LTS 日志流，但是事件列表界面不支持查看其它组织成员的审计日志。
management_event_selector	<b>ManagementEventSelector</b> object	管理类事件选择器。
agency_name	String	云服务委托名称。

表 6-47 Lts

参数	参数类型	描述
is_lts_enabled	Boolean	是否启用日志服务检索功能。
log_group_name	String	云审计服务在日志服务中创建的日志组名称。
log_topic_name	String	云审计服务在日志服务中创建的日志主题名称。

表 6-48 ObsInfo

参数	参数类型	描述
bucket_name	String	标识 OBS 桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为 3~63 个字符。
file_prefix_name	String	标识需要存储于 OBS 的日志文件前缀，0-9, a-z, A-Z, '-', '.', '_' 长度为 0~64 字符。
is_obs_created	Boolean	标识配置桶是否由追踪器自动创建。

参数	参数类型	描述
is_authorized_bucket	Boolean	标识配置桶是否已经授权给CTS服务账号。
bucket_lifecycle	Long	标识配置桶内对象存储周期。当"tracker_type"参数值为"data"时该参数值有效。
compress_type	String	压缩类型。包括不压缩 ( json )，压缩 ( gzip ) 两种状态。默认为gzip格式。 枚举值： <ul style="list-style-type: none"> <li>• <b>gzip</b></li> <li>• <b>json</b></li> </ul>
is_sort_by_service	Boolean	路径按云服务划分，打开后转储文件路径中将增加云服务名。默认为true。

表 6-49 DataBucketQuery

参数	参数类型	描述
data_bucket_name	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。
search_enabled	Boolean	追踪桶日志是否支持搜索。
data_event	Array of strings	数据类追踪器追踪对象的桶名。 <ul style="list-style-type: none"> <li>• 当启用或者停用数据类追踪器时，该参数为必选。</li> <li>• 管理类追踪器无此参数。</li> <li>• 追踪器一旦创建追踪桶无法修改。</li> <li>• READ OBS对象读取操作；WRITE OBS对象写操作。</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>WRITE</b></li> <li>• <b>READ</b></li> </ul>

表 6-50 ManagementEventSelector

参数	参数类型	描述
exclude_service	Array of strings	标识不转储的云服务名称。目前只支持设置为KMS，表示屏蔽KMS服务的createDatakey事件。

**状态码： 400**

**表 6-51** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 401**

**表 6-52** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 403**

**表 6-53** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 500**

**表 6-54** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 503**

**表 6-55** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。

参数	参数类型	描述
error_msg	String	错误描述。

## 请求示例

```
GET https://{endpoint}/v3/{project_id}/trackers?tracker_name=system
```

## 响应示例

状态码： 200

请求成功。

```
{
  "trackers": [ {
    "is_support_trace_files_encryption": true,
    "create_time": 1589886034121,
    "stream_id": "4a1ef2b6-d79a-4dc6-90f0-48151cd5491b",
    "kms_id": "7dbbb3fa-93e4-4528-bc7b-9beb794b0229",
    "group_id": "26fa12ac-75f7-42ed-8118-ab9f2263042f",
    "agency_name": "cts_admin_trust",
    "is_support_validate": false,
    "obs_info": {
      "is_obs_created": false,
      "bucket_name": "",
      "is_authorized_bucket": false,
      "file_prefix_name": "",
      "bucket_lifecycle": 0
    },
    "lts": {
      "log_group_name": "CTS",
      "is_lts_enabled": true,
      "log_topic_name": "system-trace"
    },
    "tracker_type": "system",
    "domain_id": "2306579dc99f4c8690b14b68e734fcd9",
    "project_id": "24edf66e79d04187acb99a463e610764",
    "tracker_name": "system",
    "id": "ebf8d1c3-762b-4ce3-b316-6b1aa32f8be3",
    "status": "enabled"
  }, {
    "domain_id": "2306579dc99f4c8690b14b68e734fcd9",
    "is_support_trace_files_encryption": false,
    "obs_info": {
      "is_obs_created": false,
      "bucket_name": "",
      "is_authorized_bucket": false,
      "file_prefix_name": "",
      "bucket_lifecycle": 0
    },
    "create_time": 1589276171198,
    "project_id": "24edf66e79d04187acb99a463e610764",
    "data_bucket": {
      "data_event": [ "READ", "WRITE" ],
      "search_enabled": false,
      "data_bucket_name": "ctest0423"
    },
    "tracker_name": "sdsa",
    "is_support_validate": false,
    "lts": {
      "log_group_name": "CTS",
      "is_lts_enabled": false,
      "log_topic_name": "sdsa"
    }
  },
  ]
}
```

```
"id" : "c9a3961d-3aa0-4e60-8e63-dd4ce7f1a88a",  
"status" : "enabled",  
"tracker_type" : "data"  
}]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;  
import com.huaweicloud.sdk.cts.v3.*;  
import com.huaweicloud.sdk.cts.v3.model.*;  
  
public class ListTrackersSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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 BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        CtsClient client = CtsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListTrackersRequest request = new ListTrackersRequest();  
        request.withTrackerName("<tracker_name>");  
        request.withTrackerType(ListTrackersRequest.TrackerTypeEnum.fromValue("<tracker_type>"));  
        try {  
            ListTrackersResponse response = client.listTrackers(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

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

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListTrackersRequest()
        request.tracker_name = "<tracker_name>"
        request.tracker_type = "<tracker_type>"
        response = client.list_trackers(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTrackersRequest{}
    trackerNameRequest := "<tracker_name>"
    request.TrackerName = &trackerNameRequest
```

```
trackerTypeRequest:= model.GetListTrackersRequestTrackerTypeEnum().<TRACKER_TYPE>
request.TrackerType = &trackerTypeRequest
response, err := client.ListTrackers(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.2.4 删除追踪器

### 功能介绍

云审计服务目前仅支持删除已创建的数据类追踪器。删除追踪器对已有的操作记录没有影响，当您重新开通云审计服务后，依旧可以查看已有的操作记录。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{project\_id}/trackers

表 6-56 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

表 6-57 Query 参数

参数	是否必选	参数类型	描述
tracker_name	否	String	标识追踪器名称。在不传入该字段的情况下, 将删除当前租户所有的数据类追踪器。
tracker_type	否	String	标识追踪器类型。目前仅支持数据类追踪器 (data) 的删除, 默认值为"data"。 枚举值: <ul style="list-style-type: none"><li>• data</li></ul>

## 请求参数

无

## 响应参数

状态码: 400

表 6-58 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 401

表 6-59 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。



**状态码： 403**

**表 6-60** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 404**

**表 6-61** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 500**

**表 6-62** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

**状态码： 503**

**表 6-63** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

```
DELETE https://{endpoint}/v3/{project_id}/trackers?tracker_name=data-tracker-name
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

public class DeleteTrackerSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteTrackerRequest request = new DeleteTrackerRequest();
        request.withTrackerName("<tracker_name>");
        request.withTrackerType(DeleteTrackerRequest.TrackerTypeEnum.fromValue("<tracker_type>"));
        try {
            DeleteTrackerResponse response = client.deleteTracker(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

if __name__ == "__main__":
```

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

credentials = BasicCredentials(ak, sk)

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

try:
    request = DeleteTrackerRequest()
    request.tracker_name = "<tracker_name>"
    request.tracker_type = "<tracker_type>"
    response = client.delete_tracker(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteTrackerRequest{}
    trackerNameRequest := "<tracker_name>"
    request.TrackerName = &trackerNameRequest
    trackerTypeRequest := model.GetDeleteTrackerRequestTrackerTypeEnum().<TRACKER_TYPE>
    request.TrackerType = &trackerTypeRequest
    response, err := client.DeleteTracker(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

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

## 状态码

状态码	描述
204	删除成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	服务器无法找到被请求的资源或部分追踪器删除失败。
500	服务内部异常，请求未完成；或部分追踪器删除失败。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.3 关键操作通知管理

### 6.3.1 创建关键操作通知

#### 功能介绍

配置关键操作通知，可在发生特定操作时，使用预先创建好的SMN主题，向用户手机、邮箱发送消息，也可直接发送http/https消息。常用于实时感知高危操作、触发特定操作或对接用户自有审计分析系统。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/notifications

表 6-64 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

## 请求参数

表 6-65 请求 Body 参数

参数	是否必选	参数类型	描述
notification_name	是	String	标识关键操作名称。
operation_type	是	String	标识操作类型。目前支持的操作类型有完整类型(complete)和自定义类型(customized)。完整类型下, CTS发送通知的对象为已对接服务的所有事件, 此时不用指定operations和notify_user_list字段。自定义类型下, CTS发送通知的对象是在operations列表中指定的事件。 枚举值: <ul style="list-style-type: none"><li>● complete</li><li>● customized</li></ul>
operations	否	Array of <b>Operations</b> objects	操作事件列表。
notify_user_list	否	Array of <b>NotificationUsers</b> objects	通知用户列表, 目前最多支持对10个用户组和50个用户发起的操作进行配置。
topic_id	否	String	消息通知服务的topic_urn或者函数工作流的func_urn。- 消息通知服务的topic_urn可以通过消息通知服务的查询主题列表API获取, 示例: urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2。- 函数工作流的func_urn可以通过函数工作流的获取函数列表API获取, 示例: urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test。

参数	是否必选	参数类型	描述
filter	否	<b>Filter</b> object	关键操作通知高级过滤条件。
agency_name	否	String	云服务委托名称。参数值为 "cts_admin_trust" 时，创建关键操作通知时会自动创建云服务委托：cts_admin_trust。 枚举值： <ul style="list-style-type: none"><li>• <b>cts_admin_trust</b></li></ul>

表 6-66 Operations

参数	是否必选	参数类型	描述
service_type	是	String	标识云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以查看到该云服务的英文缩写。
resource_type	是	String	标识资源类型。
trace_names	是	Array of strings	标识事件名称。

表 6-67 NotificationUsers

参数	是否必选	参数类型	描述
user_group	是	String	IAM用户组。
user_list	是	Array of strings	IAM用户。

表 6-68 Filter

参数	是否必选	参数类型	描述
condition	是	String	多条件关系。 <ul style="list-style-type: none"><li>• AND(默认值) 表示所有过滤条件满足后生效。</li><li>• OR 表示有任意一个条件满足时生效。</li></ul> 枚举值: <ul style="list-style-type: none"><li>• AND(默认值)</li><li>• OR</li></ul>
is_support_filter	是	Boolean	是否打开高级筛选开关。
rule	是	Array of strings	高级过滤条件规则, 示例如下: "key != value", 格式为: 字段规则 值。-字段取值范围: api_version,code,trace_rating,trace_type,resource_id,resource_name。-规则: != 或 =。- 值: api_version正则约束: ^(a-zA-Z0-9_-){1,64}\$; code: 最小长度1, 最大长度256; trace_rating枚举值: "normal", "warning", "incident"; trace_type枚举值: "ConsoleAction", "ApiCall", "SystemAction"; resource_id: 最小长度1, 最大长度350; resource_name: 最小长度1, 最大长度256

## 响应参数

状态码: 201

表 6-69 响应 Body 参数

参数	参数类型	描述
notification_name	String	通知名称。

参数	参数类型	描述
operation_type	String	操作类型。和自定义。 <ul style="list-style-type: none"> <li>complete: 完整类型, 对所有已对接云审计服务的所有操作发送SMN通知。</li> <li>customized: 自定义类型, 对指定云服务的指定操作发送SMN通知。</li> </ul> 枚举值: <ul style="list-style-type: none"> <li>customized</li> <li>complete</li> </ul>
operations	Array of <b>Operations</b> objects	操作事件列表。
notify_user_list	Array of <b>NotificationUsers</b> objects	通知用户列表, 目前最多支持对10个用户组和50个用户发起的操作进行配置。
status	String	通知状态。启用和停用。 <ul style="list-style-type: none"> <li>disabled: 停用关键操作通知。</li> <li>enabled: 启用关键操作通知。</li> </ul> 枚举值: <ul style="list-style-type: none"> <li>enabled</li> <li>disabled</li> </ul>
topic_id	String	消息通知服务(SMN)主题的唯一资源标识, 可通过查询主题列表获取该标识。
notification_id	String	通知的唯一标识ID。
notification_type	String	通知类型。-smn: 消息通知服务。-fun: 函数工作流。 枚举值: <ul style="list-style-type: none"> <li>smn</li> <li>fun</li> </ul>
project_id	String	项目ID。
create_time	Long	通知规则创建时间。
filter	<b>Filter</b> object	关键操作通知高级筛选条件。
agency_name	String	云服务委托名称。参数值为"cts_admin_trust"时, 创建关键操作通知时会自动创建云服务委托: cts_admin_trust。 枚举值: <ul style="list-style-type: none"> <li>cts_admin_trust</li> </ul>



表 6-70 Operations

参数	参数类型	描述
service_type	String	标识云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以查看到该云服务的英文缩写。
resource_type	String	标识资源类型。
trace_names	Array of strings	标识事件名称。

表 6-71 NotificationUsers

参数	参数类型	描述
user_group	String	IAM用户组。
user_list	Array of strings	IAM用户。

表 6-72 Filter

参数	参数类型	描述
condition	String	多条件关系。 <ul style="list-style-type: none"><li>• AND(默认值) 表示所有过滤条件满足后生效。</li><li>• OR 表示有任意一个条件满足时生效。</li></ul> 枚举值： <ul style="list-style-type: none"><li>• <b>AND(默认值)</b></li><li>• <b>OR</b></li></ul>
is_support_filter	Boolean	是否打开高级筛选开关。
rule	Array of strings	高级过滤条件规则，示例如下："key != value"，格式为：字段 规则 值。-字段取值范围：api_version,code,trace_rating,trace_type,resource_id,resource_name。-规则：!= 或 =。-值：api_version正则约束：^(a-zA-Z0-9_-){1,64}\$；code：最小长度1，最大长度256；trace_rating枚举值："normal","warning","incident"；trace_type枚举值："ConsoleAction","ApiCall","SystemAction"；resource_id：最小长度1，最大长度350；resource_name：最小长度1，最大长度256

状态码： 400

表 6-73 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 401

表 6-74 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 403

表 6-75 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 404

表 6-76 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 500

表 6-77 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 503

表 6-78 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

- 创建完整类型关键操作通知请求样例。

POST https://{endpoint}/v3/{project\_id}/notifications

```
{
  "notification_name": "test",
  "filter": {
    "is_support_filter": true,
    "rule": [ "code != 200", "api_version = v1.0", "trace_rating = normal", "trace_type != ApiCall",
"resource_id = xxx", "resource_name = xxx" ],
    "condition": "OR"
  },
  "operation_type": "complete",
  "agency_name": "cts_admin_trust",
  "topic_id": "urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:test"
}
```

- 创建自定义类型关键操作通知请求样例。

POST https://{endpoint}/v3/{project\_id}/notifications

```
{
  "notification_name": "test",
  "operation_type": "customized",
  "agency_name": "cts_admin_trust",
  "filter": {
    "is_support_filter": true,
    "rule": [ "code != 200", "api_version = v1.0", "trace_rating = normal", "trace_type != ApiCall",
"resource_id = xxx", "resource_name = xxx" ],
    "condition": "OR"
  },
  "operations": [ {
    "service_type": "CTS",
    "resource_type": "tracker",
    "trace_names": [ "createTracker", "deleteTracker" ]
  }, {
    "service_type": "CTS",
    "resource_type": "notification",
    "trace_names": [ "deleteNotification", "updateNotification" ]
  }, {
    "service_type": "AOM",
    "resource_type": "pe",
  }
]
```

```
"trace_names": [ "deletePolicyGroup", "updatePolicyGroup", "createPolicyGroup" ]
}],
"notify_user_list": [ {
  "user_group": "admin",
  "user_list": [ "test1", "test2" ]
}, {
  "user_group": "CTS view",
  "user_list": [ "test3", "test4" ]
}],
"topic_id": "urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:test"
}
```

## 响应示例

**状态码： 201**

创建成功。

```
{
  "create_time": 1634001495876,
  "notification_id": "cda8fd83-d08c-46f0-b914-1453a6a85c00",
  "notification_name": "test",
  "agency_name": "cts_admin_trust",
  "notification_type": "smn",
  "notify_user_list": [ {
    "user_group": "admin",
    "user_list": [ "test1", "test2" ]
  }, {
    "user_group": "CTS view",
    "user_list": [ "test3", "test4" ]
  } ],
  "operation_type": "customized",
  "operations": [ {
    "resource_type": "tracker",
    "service_type": "CTS",
    "trace_names": [ "createTracker", "deleteTracker" ]
  }, {
    "resource_type": "notification",
    "service_type": "CTS",
    "trace_names": [ "deleteNotification", "updateNotification" ]
  }, {
    "resource_type": "pe",
    "service_type": "AOM",
    "trace_names": [ "deletePolicyGroup", "updatePolicyGroup", "createPolicyGroup" ]
  } ],
  "project_id": "24edf66e79d04187acb99a463e610764",
  "status": "enabled",
  "topic_id": "urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:test"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 创建完整类型关键操作通知请求样例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
```

```
import com.huaweicloud.sdk.cts.v3.model.*;

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

public class CreateNotificationSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateNotificationRequest request = new CreateNotificationRequest();
        CreateNotificationRequestBody body = new CreateNotificationRequestBody();
        List<String> listFilterRule = new ArrayList<>();
        listFilterRule.add("code != 200");
        listFilterRule.add("api_version = v1.0");
        listFilterRule.add("trace_rating = normal");
        listFilterRule.add("trace_type != ApiCall");
        listFilterRule.add("resource_id = xxx");
        listFilterRule.add("resource_name = xxx");
        Filter filterbody = new Filter();
        filterbody.withCondition(Filter.ConditionEnum.fromValue("OR"))
            .withIsSupportFilter(true)
            .withRule(listFilterRule);
        body.withFilter(filterbody);
        body.withTopicId("urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:test");

        body.withAgencyName(CreateNotificationRequestBody.AgencyNameEnum.fromValue("cts_admin_trust"));

        body.withOperationType(CreateNotificationRequestBody.OperationTypeEnum.fromValue("complete"));
        body.withNotificationName("test");
        request.withBody(body);
        try {
            CreateNotificationResponse response = client.createNotification(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 创建自定义类型关键操作通知请求样例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

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

public class CreateNotificationSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateNotificationRequest request = new CreateNotificationRequest();
        CreateNotificationRequestBody body = new CreateNotificationRequestBody();
        List<String> listFilterRule = new ArrayList<>();
        listFilterRule.add("code != 200");
        listFilterRule.add("api_version = v1.0");
        listFilterRule.add("trace_rating = normal");
        listFilterRule.add("trace_type != ApiCall");
        listFilterRule.add("resource_id = xxx");
        listFilterRule.add("resource_name = xxx");
        Filter filterbody = new Filter();
        filterbody.withCondition(Filter.ConditionEnum.fromValue("OR"))
            .withIsSupportFilter(true)
            .withRule(listFilterRule);
        List<String> listNotifyUserListUserList = new ArrayList<>();
        listNotifyUserListUserList.add("test3");
        listNotifyUserListUserList.add("test4");
        List<String> listNotifyUserListUserList1 = new ArrayList<>();
        listNotifyUserListUserList1.add("test1");
        listNotifyUserListUserList1.add("test2");
        List<NotificationUsers> listbodyNotifyUserList = new ArrayList<>();
        listbodyNotifyUserList.add(
            new NotificationUsers()
                .withUserGroup("admin")
                .withUserList(listNotifyUserListUserList1)
        );
        listbodyNotifyUserList.add(
            new NotificationUsers()
                .withUserGroup("CTS view")
                .withUserList(listNotifyUserListUserList)
        );
        List<String> listOperationsTraceNames = new ArrayList<>();
        listOperationsTraceNames.add("deletePolicyGroup");
        listOperationsTraceNames.add("updatePolicyGroup");
        listOperationsTraceNames.add("createPolicyGroup");
        List<String> listOperationsTraceNames1 = new ArrayList<>();
        listOperationsTraceNames1.add("deleteNotification");
        listOperationsTraceNames1.add("updateNotification");
        List<String> listOperationsTraceNames2 = new ArrayList<>();
        listOperationsTraceNames2.add("createTracker");
        listOperationsTraceNames2.add("deleteTracker");
```

```
List<Operations> listbodyOperations = new ArrayList<>();
listbodyOperations.add(
    new Operations()
        .withServiceType("CTS")
        .withResourceType("tracker")
        .withTraceNames(listOperationsTraceNames2)
);
listbodyOperations.add(
    new Operations()
        .withServiceType("CTS")
        .withResourceType("notification")
        .withTraceNames(listOperationsTraceNames1)
);
listbodyOperations.add(
    new Operations()
        .withServiceType("AOM")
        .withResourceType("pe")
        .withTraceNames(listOperationsTraceNames)
);
body.withFilter(filterbody);
body.withTopicId("urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:test");
body.withNotifyUserList(listbodyNotifyUserList);
body.withOperations(listbodyOperations);

body.withAgencyName(CreateNotificationRequestBody.AgencyNameEnum.fromValue("cts_admin_trust"));

body.withOperationType(CreateNotificationRequestBody.OperationTypeEnum.fromValue("customized"));
body.withNotificationName("test");
request.withBody(body);
try {
    CreateNotificationResponse response = client.createNotification(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

- 创建完整类型关键操作通知请求样例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

```
credentials = BasicCredentials(ak, sk)

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

try:
    request = CreateNotificationRequest()
    listRuleFilter = [
        "code != 200",
        "api_version = v1.0",
        "trace_rating = normal",
        "trace_type != ApiCall",
        "resource_id = xxx",
        "resource_name = xxx"
    ]
    filterbody = Filter(
        condition="OR",
        is_support_filter=True,
        rule=listRuleFilter
    )
    request.body = CreateNotificationRequestBody(
        filter=filterbody,
        topic_id="urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:test",
        agency_name="cts_admin_trust",
        operation_type="complete",
        notification_name="test"
    )
    response = client.create_notification(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 创建自定义类型关键操作通知请求样例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = CreateNotificationRequest()
        listRuleFilter = [
            "code != 200",
            "api_version = v1.0",
            "trace_rating = normal",
            "trace_type != ApiCall",
```



```
        "resource_id = xxx",
        "resource_name = xxx"
    ]
    filterbody = Filter(
        condition="OR",
        is_support_filter=True,
        rule=listRuleFilter
    )
    listUserListNotifyUserList = [
        "test3",
        "test4"
    ]
    listUserListNotifyUserList1 = [
        "test1",
        "test2"
    ]
    listNotifyUserListbody = [
        NotificationUsers(
            user_group="admin",
            user_list=listUserListNotifyUserList1
        ),
        NotificationUsers(
            user_group="CTS view",
            user_list=listUserListNotifyUserList
        )
    ]
    listTraceNamesOperations = [
        "deletePolicyGroup",
        "updatePolicyGroup",
        "createPolicyGroup"
    ]
    listTraceNamesOperations1 = [
        "deleteNotification",
        "updateNotification"
    ]
    listTraceNamesOperations2 = [
        "createTracker",
        "deleteTracker"
    ]
    listOperationsbody = [
        Operations(
            service_type="CTS",
            resource_type="tracker",
            trace_names=listTraceNamesOperations2
        ),
        Operations(
            service_type="CTS",
            resource_type="notification",
            trace_names=listTraceNamesOperations1
        ),
        Operations(
            service_type="AOM",
            resource_type="pe",
            trace_names=listTraceNamesOperations
        )
    ]
    request.body = CreateNotificationRequestBody(
        filter=filterbody,
        topic_id="urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:test",
        notify_user_list=listNotifyUserListbody,
        operations=listOperationsbody,
        agency_name="cts_admin_trust",
        operation_type="customized",
        notification_name="test"
    )
    response = client.create_notification(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
```

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

## Go

- 创建完整类型关键操作通知请求样例。

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNotificationRequest{}
    var listRuleFilter = []string{
        "code != 200",
        "api_version = v1.0",
        "trace_rating = normal",
        "trace_type != ApiCall",
        "resource_id = xxx",
        "resource_name = xxx",
    }
    filterbody := &model.Filter{
        Condition: model.GetFilterConditionEnum().OR,
        IsSupportFilter: true,
        Rule: listRuleFilter,
    }
    topicIdCreateNotificationRequestBody := "urn:smn:
{regionid}:24edf66e79d04187acb99a463e610764:test"
    agencyNameCreateNotificationRequestBody :=
model.GetCreateNotificationRequestBodyAgencyNameEnum().CTS_ADMIN_TRUST
    request.Body = &model.CreateNotificationRequestBody{
        Filter: filterbody,
        TopicId: &topicIdCreateNotificationRequestBody,
        AgencyName: &agencyNameCreateNotificationRequestBody,
        OperationType: model.GetCreateNotificationRequestBodyOperationTypeEnum().COMPLETE,
        NotificationName: "test",
    }
    response, err := client.CreateNotification(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 创建自定义类型关键操作通知请求样例。

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNotificationRequest{}
    var listRuleFilter = []string{
        "code != 200",
        "api_version = v1.0",
        "trace_rating = normal",
        "trace_type != ApiCall",
        "resource_id = xxx",
        "resource_name = xxx",
    }
    filterbody := &model.Filter{
        Condition: model.GetFilterConditionEnum().OR,
        IsSupportFilter: true,
        Rule: listRuleFilter,
    }
    var listUserListNotifyUserList = []string{
        "test3",
        "test4",
    }
    var listUserListNotifyUserList1 = []string{
        "test1",
        "test2",
    }
    var listNotifyUserListbody = []model.NotificationUsers{
        {
            UserGroup: "admin",
            UserList: listUserListNotifyUserList1,
        },
        {
            UserGroup: "CTS view",
            UserList: listUserListNotifyUserList,
        },
    }
    var listTraceNamesOperations = []string{
        "deletePolicyGroup",
        "updatePolicyGroup",
    }
}
```

```
        "createPolicyGroup",
    }
    var listTraceNamesOperations1 = []string{
        "deleteNotification",
        "updateNotification",
    }
    var listTraceNamesOperations2 = []string{
        "createTracker",
        "deleteTracker",
    }
    var listOperationsbody = []model.Operations{
        {
            ServiceType: "CTS",
            ResourceType: "tracker",
            TraceNames: listTraceNamesOperations2,
        },
        {
            ServiceType: "CTS",
            ResourceType: "notification",
            TraceNames: listTraceNamesOperations1,
        },
        {
            ServiceType: "AOM",
            ResourceType: "pe",
            TraceNames: listTraceNamesOperations,
        },
    }
    topicIdCreateNotificationRequestBody:= "urn:smn:
{regionid}:24edf66e79d04187acb99a463e610764:test"
    agencyNameCreateNotificationRequestBody:=
model.GetCreateNotificationRequestBodyAgencyNameEnum().CTS_ADMIN_TRUST
    request.Body = &model.CreateNotificationRequestBody{
        Filter: filterbody,
        TopicId: &topicIdCreateNotificationRequestBody,
        NotifyUserList: &listNotifyUserListbody,
        Operations: &listOperationsbody,
        AgencyName: &agencyNameCreateNotificationRequestBody,
        OperationType: model.GetCreateNotificationRequestBodyOperationTypeEnum().CUSTOMIZED,
        NotificationName: "test",
    }
    response, err := client.CreateNotification(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
201	创建成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。

状态码	描述
404	服务器无法找到被请求的资源或部分关键操作通知删除失败。
500	服务内部异常，请求未完成；或部分追踪器删除失败。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.3.2 修改关键操作通知

### 功能介绍

云审计服务支持修改已创建关键操作通知配置项，通过notification\_id的字段匹配修改对象，notification\_id必须已经存在。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/notifications

表 6-79 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，参见 <a href="#">获取账号ID和项目ID</a> 章节。

### 请求参数

表 6-80 请求 Body 参数

参数	是否必选	参数类型	描述
notification_name	是	String	标识关键操作名称。

参数	是否必选	参数类型	描述
operation_type	是	String	标识操作类型。目前支持的操作类型有完整类型(complete)和自定义类型(customized)。完整类型下, CTS发送通知的对象为已对接服务的所有事件。自定义类型下, CTS发送通知的对象是在operations列表中指定的事件。 枚举值: <ul style="list-style-type: none"> <li>• <b>customized</b></li> <li>• <b>complete</b></li> </ul>
operations	否	Array of <b>Operations</b> objects	操作事件列表。
notify_user_list	否	Array of <b>NotificationUsers</b> objects	通知用户列表, 目前最多支持对10个用户组和50个用户发起的操作进行配置。
status	是	String	标识关键操作通知状态, 包括正常(enabled), 停止(disabled)两种状态。 枚举值: <ul style="list-style-type: none"> <li>• <b>enabled</b></li> <li>• <b>disabled</b></li> </ul>
topic_id	否	String	消息通知服务的topic_urn或者函数工作流的func_urn, 当“status”字段为enabled时, 该字段必填。- 消息通知服务的topic_urn可以通过消息通知服务的查询主题列表API获取, 示例: urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2。- 函数工作流的func_urn可以通过函数工作流的获取函数列表API获取, 示例: urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test。
notification_id	是	String	关键操作通知id。
filter	否	<b>Filter</b> object	关键操作通知高级筛选条件。

参数	是否必选	参数类型	描述
agency_name	否	String	云服务委托名称。参数值为 "cts_admin_trust" 时，创建关键操作通知时会自动创建云服务委托：cts_admin_trust。 枚举值： <ul style="list-style-type: none"><li>• cts_admin_trust</li></ul>

表 6-81 Operations

参数	是否必选	参数类型	描述
service_type	是	String	标识云服务类型。必须为已对接 CTS 的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以查看到该云服务的英文缩写。
resource_type	是	String	标识资源类型。
trace_names	是	Array of strings	标识事件名称。

表 6-82 NotificationUsers

参数	是否必选	参数类型	描述
user_group	是	String	IAM 用户组。
user_list	是	Array of strings	IAM 用户。

表 6-83 Filter

参数	是否必选	参数类型	描述
condition	是	String	多条件关系。 <ul style="list-style-type: none"><li>• AND(默认值) 表示所有过滤条件满足后生效。</li><li>• OR 表示有任意一个条件满足时生效。</li></ul> 枚举值: <ul style="list-style-type: none"><li>• AND(默认值)</li><li>• OR</li></ul>
is_support_filter	是	Boolean	是否打开高级筛选开关。
rule	是	Array of strings	高级过滤条件规则, 示例如下: "key != value", 格式为: 字段规则 值。-字段取值范围: api_version,code,trace_rating,trace_type,resource_id,resource_name。-规则: != 或 =。- 值: api_version正则约束: ^(a-zA-Z0-9_-){1,64}\$; code: 最小长度1, 最大长度256; trace_rating枚举值: "normal", "warning", "incident"; trace_type枚举值: "ConsoleAction", "ApiCall", "SystemAction"; resource_id: 最小长度1, 最大长度350; resource_name: 最小长度1, 最大长度256

## 响应参数

状态码: 200

表 6-84 响应 Body 参数

参数	参数类型	描述
notification_name	String	标识关键操作名称。



参数	参数类型	描述
operation_type	String	标识操作类型。目前支持的操作类型有完整类型(complete)和自定义类型(customized)。完整类型下,CTS发送通知的对象为已对接服务的所有事件。自定义类型下,CTS发送通知的对象是在operations列表中指定的事件。 枚举值: <ul style="list-style-type: none"> <li>● customized</li> <li>● complete</li> </ul>
operations	Array of <b>Operations</b> objects	操作事件列表。
notify_user_list	Array of <b>NotificationUsers</b> objects	通知用户列表,目前最多支持对10个用户组和50个用户发起的操作进行配置。
status	String	标识关键操作通知状态,包括正常(enabled),停止(disabled)两种状态。 枚举值: <ul style="list-style-type: none"> <li>● enabled</li> <li>● disabled</li> </ul>
topic_id	String	消息通知服务的topic_urn或者函数工作流的func_urn。 - 消息通知服务的topic_urn可以通过消息通知服务的查询主题列表API获取,示例:urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2。 - 函数工作流的func_urn可以通过函数工作流的获取函数列表API获取,示例:urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test。
notification_id	String	关键操作通知的唯一标识。
notification_type	String	关键操作通知类型,根据topic_id区分为消息通知服务(smn)和函数工作流(fun)。 枚举值: <ul style="list-style-type: none"> <li>● smn</li> <li>● fun</li> </ul>
project_id	String	项目ID。
create_time	Long	关键操作通知创建时间戳。
filter	<b>Filter</b> object	关键操作通知高级筛选条件。

参数	参数类型	描述
agency_name	String	云服务委托名称。参数值为"cts_admin_trust"时，修改关键操作通知时会自动创建云服务委托：cts_admin_trust。 枚举值： <ul style="list-style-type: none"><li>• cts_admin_trust</li></ul>

表 6-85 Operations

参数	参数类型	描述
service_type	String	标识云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以查看到该云服务的英文缩写。
resource_type	String	标识资源类型。
trace_names	Array of strings	标识事件名称。

表 6-86 NotificationUsers

参数	参数类型	描述
user_group	String	IAM用户组。
user_list	Array of strings	IAM用户。

表 6-87 Filter

参数	参数类型	描述
condition	String	多条件关系。 <ul style="list-style-type: none"><li>• AND(默认值) 表示所有过滤条件满足后生效。</li><li>• OR 表示有任意一个条件满足时生效。</li></ul> 枚举值： <ul style="list-style-type: none"><li>• AND(默认值)</li><li>• OR</li></ul>
is_support_filter	Boolean	是否打开高级筛选开关。

参数	参数类型	描述
rule	Array of strings	高级过滤条件规则，示例如下："key != value"，格式为：字段 规则 值。-字段取值范围：api_version,code,trace_rating,trace_type,resource_id,resource_name。-规则：!= 或 =。-值：api_version正则约束：^(a-zA-Z0-9_-){1,64}\$；code：最小长度1，最大长度256；trace_rating枚举值："normal", "warning", "incident"；trace_type枚举值："ConsoleAction", "ApiCall", "SystemAction"；resource_id：最小长度1，最大长度350；resource_name：最小长度1，最大长度256

状态码：400

表 6-88 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

状态码：401

表 6-89 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

状态码：403

表 6-90 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

状态码：404

表 6-91 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 500

表 6-92 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 503

表 6-93 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

修改关键操作通知请求样例。

PUT https://{endpoint}/v3/{project\_id}/notifications

```
{
  "notification_id": "6d4a09bb-aa8e-40db-9e87-0d5e203823a8",
  "notification_name": "test",
  "agency_name": "cts_admin_trust",
  "operation_type": "customized",
  "operations": [ {
    "service_type": "CTS",
    "resource_type": "tracker",
    "trace_names": [ "createTracker", "deleteTracker" ]
  }, {
    "service_type": "CTS",
    "resource_type": "notification",
    "trace_names": [ "deleteNotification", "updateNotification" ]
  }, {
    "service_type": "AOM",
    "resource_type": "pe",
    "trace_names": [ "deletePolicyGroup", "updatePolicyGroup", "createPolicyGroup" ]
  } ],
  "notify_user_list": [ {
    "user_group": "admin",
    "user_list": [ "test", "test1" ]
  } ]
}
```

```
}, {  
  "user_group": "CTS view",  
  "user_list": [ "test2", "test3" ]  
}],  
"status": "enabled",  
"topic_id": "urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:foo"  
}
```

## 响应示例

**状态码: 200**

修改关键操作通知成功。

```
{  
  "notification_id": "6d4a09bb-aa8e-40db-9e87-0d5e203823a8",  
  "notification_name": "test",  
  "agency_name": "cts_admin_trust",  
  "operation_type": "customized",  
  "operations": [ {  
    "service_type": "CTS",  
    "resource_type": "tracker",  
    "trace_names": [ "createTracker", "deleteTracker" ]  
  }, {  
    "service_type": "CTS",  
    "resource_type": "notification",  
    "trace_names": [ "deleteNotification", "updateNotification" ]  
  }, {  
    "service_type": "AOM",  
    "resource_type": "pe",  
    "trace_names": [ "deletePolicyGroup", "updatePolicyGroup", "createPolicyGroup" ]  
  } ],  
  "notify_user_list": [ {  
    "user_group": "admin",  
    "user_list": [ "test", "test1" ]  
  }, {  
    "user_group": "CTS view",  
    "user_list": [ "test2", "test3" ]  
  } ],  
  "status": "enabled",  
  "project_id": "24edf66e79d04187acb99a463e610764",  
  "notification_type": "smn",  
  "create_time": 1634001495876,  
  "topic_id": "urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:foo"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

修改关键操作通知请求样例。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;  
import com.huaweicloud.sdk.cts.v3.*;  
import com.huaweicloud.sdk.cts.v3.model.*;  
  
import java.util.List;
```

```
import java.util.ArrayList;

public class UpdateNotificationSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateNotificationRequest request = new UpdateNotificationRequest();
        UpdateNotificationRequestBody body = new UpdateNotificationRequestBody();
        List<String> listNotifyUserListUserList = new ArrayList<>();
        listNotifyUserListUserList.add("test2");
        listNotifyUserListUserList.add("test3");
        List<String> listNotifyUserListUserList1 = new ArrayList<>();
        listNotifyUserListUserList1.add("test");
        listNotifyUserListUserList1.add("test1");
        List<NotificationUsers> listbodyNotifyUserList = new ArrayList<>();
        listbodyNotifyUserList.add(
            new NotificationUsers()
                .withUserGroup("admin")
                .withUserList(listNotifyUserListUserList1)
        );
        listbodyNotifyUserList.add(
            new NotificationUsers()
                .withUserGroup("CTS view")
                .withUserList(listNotifyUserListUserList)
        );
        List<String> listOperationsTraceNames = new ArrayList<>();
        listOperationsTraceNames.add("deletePolicyGroup");
        listOperationsTraceNames.add("updatePolicyGroup");
        listOperationsTraceNames.add("createPolicyGroup");
        List<String> listOperationsTraceNames1 = new ArrayList<>();
        listOperationsTraceNames1.add("deleteNotification");
        listOperationsTraceNames1.add("updateNotification");
        List<String> listOperationsTraceNames2 = new ArrayList<>();
        listOperationsTraceNames2.add("createTracker");
        listOperationsTraceNames2.add("deleteTracker");
        List<Operations> listbodyOperations = new ArrayList<>();
        listbodyOperations.add(
            new Operations()
                .withServiceType("CTS")
                .withResourceType("tracker")
                .withTraceNames(listOperationsTraceNames2)
        );
        listbodyOperations.add(
            new Operations()
                .withServiceType("CTS")
                .withResourceType("notification")
                .withTraceNames(listOperationsTraceNames1)
        );
        listbodyOperations.add(
            new Operations()
                .withServiceType("AOM")
                .withResourceType("pe")
                .withTraceNames(listOperationsTraceNames)
        );
    }
}
```

```
body.withNotificationId("6d4a09bb-aa8e-40db-9e87-0d5e203823a8");
body.withTopicId("urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:foo");
body.withStatus(UpdateNotificationRequestBody.StatusEnum.fromValue("enabled"));
body.withNotifyUserList(listbodyNotifyUserList);
body.withOperations(listbodyOperations);

body.withAgencyName(UpdateNotificationRequestBody.AgencyNameEnum.fromValue("cts_admin_trust"));

body.withOperationType(UpdateNotificationRequestBody.OperationTypeEnum.fromValue("customized"));
body.withNotificationName("test");
request.withBody(body);
try {
    UpdateNotificationResponse response = client.updateNotification(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

修改关键操作通知请求样例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = UpdateNotificationRequest()
        listUserListNotifyUserList = [
            "test2",
            "test3"
        ]
        listUserListNotifyUserList1 = [
            "test",
            "test1"
        ]
        listNotifyUserListbody = [
            NotificationUsers(
                user_group="admin",
```

```
        user_list=listUserListNotifyUserList1
    ),
    NotificationUsers(
        user_group="CTS view",
        user_list=listUserListNotifyUserList
    )
]
listTraceNamesOperations = [
    "deletePolicyGroup",
    "updatePolicyGroup",
    "createPolicyGroup"
]
listTraceNamesOperations1 = [
    "deleteNotification",
    "updateNotification"
]
listTraceNamesOperations2 = [
    "createTracker",
    "deleteTracker"
]
listOperationsbody = [
    Operations(
        service_type="CTS",
        resource_type="tracker",
        trace_names=listTraceNamesOperations2
    ),
    Operations(
        service_type="CTS",
        resource_type="notification",
        trace_names=listTraceNamesOperations1
    ),
    Operations(
        service_type="AOM",
        resource_type="pe",
        trace_names=listTraceNamesOperations
    )
]
request.body = UpdateNotificationRequestBody(
    notification_id="6d4a09bb-aa8e-40db-9e87-0d5e203823a8",
    topic_id="urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:foo",
    status="enabled",
    notify_user_list=listNotifyUserListbody,
    operations=listOperationsbody,
    agency_name="cts_admin_trust",
    operation_type="customized",
    notification_name="test"
)
response = client.update_notification(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

修改关键操作通知请求样例。

```
package main

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



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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNotificationRequest{}
    var userListNotifyUserList = []string{
        "test2",
        "test3",
    }
    var userListNotifyUserList1 = []string{
        "test",
        "test1",
    }
    var listNotifyUserListbody = []model.NotificationUsers{
        {
            UserGroup: "admin",
            UserList: userListNotifyUserList1,
        },
        {
            UserGroup: "CTS view",
            UserList: userListNotifyUserList,
        },
    }
    var listTraceNamesOperations = []string{
        "deletePolicyGroup",
        "updatePolicyGroup",
        "createPolicyGroup",
    }
    var listTraceNamesOperations1 = []string{
        "deleteNotification",
        "updateNotification",
    }
    var listTraceNamesOperations2 = []string{
        "createTracker",
        "deleteTracker",
    }
    var listOperationsbody = []model.Operations{
        {
            ServiceType: "CTS",
            ResourceType: "tracker",
            TraceNames: listTraceNamesOperations2,
        },
        {
            ServiceType: "CTS",
            ResourceType: "notification",
            TraceNames: listTraceNamesOperations1,
        },
        {
            ServiceType: "AOM",
            ResourceType: "pe",
            TraceNames: listTraceNamesOperations,
        },
    }
}
```

```
topicIdUpdateNotificationRequestBody:= "urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:foo"
agencyNameUpdateNotificationRequestBody:=
model.GetUpdateNotificationRequestBodyAgencyNameEnum().CTS_ADMIN_TRUST
request.Body = &model.UpdateNotificationRequestBody{
    NotificationId: "6d4a09bb-aa8e-40db-9e87-0d5e203823a8",
    TopicId: &topicIdUpdateNotificationRequestBody,
    Status: model.GetUpdateNotificationRequestBodyStatusEnum().ENABLED,
    NotifyUserList: &listNotifyUserListbody,
    Operations: &listOperationsbody,
    AgencyName: &agencyNameUpdateNotificationRequestBody,
    OperationType: model.GetUpdateNotificationRequestBodyOperationTypeEnum().CUSTOMIZED,
    NotificationName: "test",
}
response, err := client.UpdateNotification(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

## 更多

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

## 状态码

状态码	描述
200	修改关键操作通知成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	服务器无法找到被请求的资源或部分关键操作通知删除失败。
500	服务内部异常，请求未完成；或部分追踪器删除失败。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

### 6.3.3 删除关键操作通知

#### 功能介绍

云审计服务支持删除已创建的关键操作通知。

#### 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/{project\_id}/notifications

表 6-94 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

表 6-95 Query 参数

参数	是否必选	参数类型	描述
notification_id	是	String	标识关键操作通知id。批量删除请使用逗号隔开, notification_id="xxx1,cccc2"

## 请求参数

无

## 响应参数

状态码: 400

表 6-96 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 401

表 6-97 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 403

表 6-98 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 404

表 6-99 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 500

表 6-100 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 503

表 6-101 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

public class DeleteNotificationSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteNotificationRequest request = new DeleteNotificationRequest();
        request.withNotificationId("<notification_id>");
        try {
            DeleteNotificationResponse response = client.deleteNotification(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

### Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

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

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

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]

credentials = BasicCredentials(ak, sk)

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

try:
    request = DeleteNotificationRequest()
    request.notification_id = "<notification_id>"
    response = client.delete_notification(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteNotificationRequest{
        request.NotificationId = "<notification_id>"
    }
    response, err := client.DeleteNotification(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
204	删除成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	服务器无法找到被请求的资源或部分关键操作通知删除失败。
500	服务内部异常，请求未完成；或部分追踪器删除失败。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

### 6.3.4 查询关键操作通知

#### 功能介绍

查询创建的关键操作通知规则。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{project\_id}/notifications/{notification\_type}

表 6-102 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，参见 <a href="#">获取账号ID和项目ID</a> 章节。

参数	是否必选	参数类型	描述
notification_type	是	String	通知类型。-smn: 消息通知服务。-fun: 函数工作流。 枚举值: <ul style="list-style-type: none"> <li>• smn</li> <li>• fun</li> </ul>

表 6-103 Query 参数

参数	是否必选	参数类型	描述
notification_name	否	String	标识关键操作通知名称。在不传入该字段的情况下, 将查询当前租户所有的关键操作通知。

## 请求参数

无

## 响应参数

状态码: 200

表 6-104 响应 Body 参数

参数	参数类型	描述
notifications	Array of <a href="#">NotificationsResponseBody</a> objects	关键操作通知列表。

表 6-105 NotificationsResponseBody

参数	参数类型	描述
notification_name	String	标识关键操作名称。



参数	参数类型	描述
operation_type	String	标识操作类型。目前支持的操作类型有完整类型(complete)和自定义类型(customized)。完整类型下,CTS发送通知的对象为已对接服务的所有事件。自定义类型下,CTS发送通知的对象是在operations列表中指定的事件。 枚举值: <ul style="list-style-type: none"> <li>● <b>customized</b></li> <li>● <b>complete</b></li> </ul>
operations	Array of <b>Operations</b> objects	操作事件列表。
notify_user_list	Array of <b>NotificationUsers</b> objects	通知用户列表,目前最多支持对10个用户组和50个用户发起的操作进行配置。
status	String	标识关键操作通知状态,包括正常(enabled),停止(disabled)两种状态。 枚举值: <ul style="list-style-type: none"> <li>● <b>enabled</b></li> <li>● <b>disabled</b></li> </ul>
topic_id	String	消息通知服务的topic_urn或者函数工作流的func_urn。- 消息通知服务的topic_urn可以通过消息通知服务的查询主题列表API获取,示例:urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2。- 函数工作流的func_urn可以通过函数工作流的获取函数列表API获取,示例:urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test。
notification_id	String	关键操作通知的唯一标识。
notification_type	String	关键操作通知类型,根据topic_id区分为消息通知服务(smn)和函数工作流(fun)。 枚举值: <ul style="list-style-type: none"> <li>● <b>smn</b></li> <li>● <b>fun</b></li> </ul>
project_id	String	项目ID。
create_time	Long	关键操作通知创建时间戳。
filter	<b>Filter</b> object	关键操作通知高级筛选条件。
agency_name	String	云服务委托名称。

表 6-106 Operations

参数	参数类型	描述
service_type	String	标识云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以查看到该云服务的英文缩写。
resource_type	String	标识资源类型。
trace_names	Array of strings	标识事件名称。

表 6-107 NotificationUsers

参数	参数类型	描述
user_group	String	IAM用户组。
user_list	Array of strings	IAM用户。

表 6-108 Filter

参数	参数类型	描述
condition	String	多条件关系。 <ul style="list-style-type: none"><li>• AND(默认值) 表示所有过滤条件满足后生效。</li><li>• OR 表示有任意一个条件满足时生效。</li></ul> 枚举值： <ul style="list-style-type: none"><li>• <b>AND(默认值)</b></li><li>• <b>OR</b></li></ul>
is_support_filter	Boolean	是否打开高级筛选开关。
rule	Array of strings	高级过滤条件规则，示例如下："key != value"，格式为：字段 规则 值。-字段取值范围：api_version,code,trace_rating,trace_type,resource_id,resource_name。-规则：!= 或 =。-值：api_version正则约束：^(a-zA-Z0-9_-){1,64}\$；code：最小长度1，最大长度256；trace_rating枚举值："normal","warning","incident"；trace_type枚举值："ConsoleAction","ApiCall","SystemAction"；resource_id：最小长度1，最大长度350；resource_name：最小长度1，最大长度256

状态码： 400

表 6-109 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 401

表 6-110 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 403

表 6-111 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 404

表 6-112 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 500

表 6-113 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 503

表 6-114 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

无

## 响应示例

状态码: 200

查询成功。

```
{
  "notifications": [ {
    "create_time": 1633933167385,
    "notify_user_list": [ {
      "user_group": "admin",
      "user_list": [ "test1", "test2" ]
    }, {
      "user_group": "CTS view",
      "user_list": [ "test3", "test4" ]
    } ],
    "notification_id": "0b98e1c2-2fd6-4e33-a355-f9e12eaab88a",
    "notification_name": "test2",
    "notification_type": "smn",
    "operation_type": "customized",
    "operations": [ {
      "resource_type": "tracker",
      "service_type": "CTS",
      "trace_names": [ "createTracker" ]
    }, {
      "resource_type": "notification",
      "service_type": "CTS",
      "trace_names": [ "deleteNotification", "updateNotification" ]
    }, {
      "resource_type": "pe",
      "service_type": "AOM",
      "trace_names": [ "createPolicyGroup", "updatePolicyGroup", "deletePolicyGroup" ]
    } ],
    "project_id": "24edf66e79d04187acb99a463e610764",
    "status": "enabled",
    "topic_id": "urn:smn:{regionid}:24edf66e79d04187acb99a463e610764:test"
  }, {
```

```
"create_time": 1633924057706,
"notify_user_list": [ {
  "user_group": "admin",
  "user_list": [ "test1", "test2" ]
}, {
  "user_group": "CTS view",
  "user_list": [ "test3", "test4" ]
} ],
"notification_id": "6d4a09bb-aa8e-40db-9e87-0d5e203823a8",
"notification_name": "test1",
"notification_type": "smn",
"operation_type": "complete",
"operations": [ ],
"project_id": "24edf66e79d04187acb99a463e610764",
"status": "disabled"
} ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

public class ListNotificationsSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListNotificationsRequest request = new ListNotificationsRequest();
        request.withNotificationName("<notification_name>");
        try {
            ListNotificationsResponse response = client.listNotifications(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
        }
    }
}
```

```
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListNotificationsRequest()
        request.notification_name = "<notification_name>"
        response = client.list_notifications(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

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

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

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

```
client := cts.NewCtsClient(
    cts.CtsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListNotificationsRequest{
    notificationNameRequest:= "<notification_name>"
    request.NotificationName = &notificationNameRequest
    response, err := client.ListNotifications(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	查询成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	服务器无法找到被请求的资源或部分关键操作通知删除失败。
500	服务内部异常，请求未完成；或部分追踪器删除失败。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.4 其它接口

### 6.4.1 查询租户追踪器配额信息

#### 功能介绍

查询租户追踪器配额信息。

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{project\_id}/quotas

表 6-115 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

## 请求参数

无

## 响应参数

状态码: 200

表 6-116 响应 Body 参数

参数	参数类型	描述
resources	Array of <a href="#">Quota</a> objects	本次查询追踪器列表返回的追踪器数组。

表 6-117 Quota

参数	参数类型	描述
type	String	quota资源类型。
used	Long	已使用的资源个数。
quota	Long	总资源个数。

状态码: 400

表 6-118 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 401



表 6-119 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 403

表 6-120 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 404

表 6-121 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 500

表 6-122 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 503

表 6-123 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

```
GET https://{endpoint}/v3/{project_id}/quotas
```

## 响应示例

**状态码： 200**

请求成功。

```
{
  "resources" : [ {
    "type" : "data_tracker",
    "used" : 9,
    "quota" : 100
  }, {
    "type" : "system_tracker",
    "used" : 1,
    "quota" : 1
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

public class ListQuotasSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListQuotasRequest request = new ListQuotasRequest();
        try {
            ListQuotasResponse response = client.listQuotas(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

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

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

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

## Go

```
package main

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

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

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

```
WithSk(sk).
Build()

client := cts.NewCtsClient(
    cts.CtsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

## 更多

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

## 状态码

状态码	描述
200	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求资源不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.4.2 查询云服务的全量操作列表

### 功能介绍

查询云服务的全量操作列表。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{project\_id}/operations

表 6-124 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

表 6-125 Query 参数

参数	是否必选	参数类型	描述
service_type	否	String	事件对应的云服务类型。
resource_type	否	String	事件对应的资源类型。传入该参数时, service_type必选。

## 请求参数

无

## 响应参数

状态码: 200

表 6-126 响应 Body 参数

参数	参数类型	描述
operations	Array of <a href="#">ListOperation</a> objects	全量云服务的操作事件列表。

表 6-127 ListOperation

参数	参数类型	描述
service_type	String	事件对应的云服务类型。
resource_type	String	事件对应的资源类型。
operation_list	Array of strings	操作事件名称数组。

状态码: 400

表 6-128 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 401

表 6-129 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 403

表 6-130 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 404

表 6-131 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 500

表 6-132 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 503

表 6-133 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

## 请求示例

GET https://{endpoint}/v3/{project\_id}/operations

## 响应示例

状态码： 200

请求成功。

```
{
  "operations" : [ {
    "service_type" : "cts",
    "resource_type" : "tracker",
    "operation_list" : [ "updateTracker", "createTracker", "deleteTracker" ]
  }, {
    "service_type" : "cts",
    "resource_type" : "notification",
    "operation_list" : [ "updateNotification", "createNotification", "deleteNotification" ]
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

public class ListOperationsSolution {

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

```
        .withAk(ak)
        .withSk(sk);

    CtsClient client = CtsClient.newBuilder()
        .withCredential(auth)
        .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
        .build();
    ListOperationsRequest request = new ListOperationsRequest();
    request.withServiceType("<service_type>");
    request.withResourceType("<resource_type>");
    try {
        ListOperationsResponse response = client.listOperations(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListOperationsRequest()
        request.service_type = "<service_type>"
        request.resource_type = "<resource_type>"
        response = client.list_operations(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main
```



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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListOperationsRequest{}
    serviceTypeRequest := "<service_type>"
    request.ServiceType = &serviceTypeRequest
    resourceTypeRequest := "<resource_type>"
    request.ResourceType = &resourceTypeRequest
    response, err := client.ListOperations(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求资源不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.4.3 查询审计日志的操作用户列表

### 功能介绍

查询审计日志的操作用户列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/user-resources

表 6-134 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

### 请求参数

无

### 响应参数

状态码: 200

表 6-135 响应 Body 参数

参数	参数类型	描述
users	Array of <a href="#">UserResource</a> objects	最近30天操作事件的用户列表。

表 6-136 UserResource

参数	参数类型	描述
name	String	用户名。

状态码: 400

表 6-137 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 401

表 6-138 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 403

表 6-139 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 404

表 6-140 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 500

表 6-141 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 503

表 6-142 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

## 请求示例

```
GET https://{endpoint}/v3/{project_id}/user-resource
```

## 响应示例

状态码： 200

请求成功。

```
{
  "users" : [ {
    "name" : "user1"
  }, {
    "name" : "user2"
  } ]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

public class ListUserResourcesSolution {

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

        CtsClient client = CtsClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
        .build();
ListUserResourcesRequest request = new ListUserResourcesRequest();
try {
    ListUserResourcesResponse response = client.listUserResources(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

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

## Go

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

## 更多

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

## 状态码

状态码	描述
200	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求资源不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.4.4 检查已经配置 OBS 桶是否可以成功转储

### 功能介绍

检查已经配置OBS桶是否可以成功转储。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{domain\_id}/checkbucket

表 6-143 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户id, 参见 <a href="#">获取账号ID和项目ID</a> 章节。

### 请求参数

表 6-144 请求 Body 参数

参数	是否必选	参数类型	描述
buckets	否	Array of <a href="#">CheckBucketRequest</a> objects	请求检查的OBS桶列表。

表 6-145 CheckBucketRequest

参数	是否必选	参数类型	描述
bucket_name	是	String	标识OBS桶名称。由数字或字母开头, 支持小写字母、数字、“-”、“.”, 长度为3~63个字符。
bucket_location	是	String	标识OBS桶位置。
kms_id	否	String	事件文件转储加密所采用的密钥id, <a href="#">is_support_trace_files_encryption</a> 参数值为“是”时, 此参数为必选项。

参数	是否必选	参数类型	描述
is_support_trace_files_encryption	否	Boolean	事件文件转储加密功能开关。该参数必须与kms_id参数同时使用。

## 响应参数

状态码： 200

表 6-146 响应 Body 参数

参数	参数类型	描述
buckets	Array of <b>Bucket</b> objects	检查OBS桶状态响应体。

表 6-147 Bucket

参数	参数类型	描述
bucket_name	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。
bucket_location	String	标识桶位置。
kms_id	String	事件文件转储加密所采用的密钥id。如果is_support_trace_files_encryption"参数值为“是”时，此参数为必选项。
is_support_trace_files_encryption	Boolean	事件文件转储加密功能开关。该参数必须与kms_id参数同时使用。
check_bucket_response	<b>CheckBucketResponse</b> object	转储OBS桶的检查结果。

表 6-148 CheckBucketResponse

参数	参数类型	描述
error_code	String	错误码。
error_message	String	错误信息。



参数	参数类型	描述
response_code	Integer	返回的http状态码。
success	Boolean	是否成功转储。

状态码： 400

表 6-149 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 401

表 6-150 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 403

表 6-151 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 404

表 6-152 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 500

表 6-153 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 503

表 6-154 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

检查obs桶状态请求体示例

POST https://{endpoint}/v3/{domain\_id}/checkbucket

```
{
  "buckets": [ {
    "bucket_location": "cn-north-1",
    "bucket_name": "bucket1",
    "is_support_trace_files_encryption": false,
    "kms_id": "1f26f8d8-65d4-436b-bea2-bd0ac1984f71"
  }, {
    "bucket_location": "cn-north-2",
    "bucket_name": "bucket2",
    "is_support_trace_files_encryption": false,
    "kms_id": "0c1b7d87-5186-411a-86ce-ed3b2ec848c9"
  } ]
}
```

## 响应示例

状态码： 200

请求成功。

```
{
  "buckets": [ {
    "bucket_location": "cn-north-1",
    "bucket_name": "bucket1",
    "check_bucket_response": {
      "response_code": 200,
      "success": true
    },
    "is_support_trace_files_encryption": false,
    "kms_id": "1f26f8d8-65d4-436b-bea2-bd0ac1984f71"
  }, {
    "bucket_location": "cn-north-2",
```

```
"bucket_name": "bucket2",
"check_bucket_response": {
  "error_code": "OBS.NoSuchBucket",
  "error_message": "Error message:Request Error.OBS service Error Message.",
  "response_code": 404,
  "success": false
},
"is_support_trace_files_encryption": false,
"kms_id": "0c1b7d87-5186-411a-86ce-ed3b2ec848c9"
}]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

#### 检查obs桶状态请求体示例

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

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

public class CheckObsBucketsSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        CheckObsBucketsRequest request = new CheckObsBucketsRequest();
        CheckObsBucketsRequestBody body = new CheckObsBucketsRequestBody();
        List<CheckBucketRequest> listbodyBuckets = new ArrayList<>();
        listbodyBuckets.add(
            new CheckBucketRequest()
                .withBucketName("bucket1")
                .withBucketLocation("cn-north-1")
                .withKmsId("1f26f8d8-65d4-436b-bea2-bd0ac1984f71")
                .withIsSupportTraceFilesEncryption(false)
        );
        listbodyBuckets.add(
            new CheckBucketRequest()
                .withBucketName("bucket2")
                .withBucketLocation("cn-north-2")
        );
    }
}
```

```
        .withKmsId("0c1b7d87-5186-411a-86ce-ed3b2ec848c9")
        .withIsSupportTraceFilesEncryption(false)
    );
    body.withBuckets(listbodyBuckets);
    request.withBody(body);
    try {
        CheckObsBucketsResponse response = client.checkObsBuckets(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

### 检查obs桶状态请求体示例

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = CheckObsBucketsRequest()
        listBucketsbody = [
            CheckBucketRequest(
                bucket_name="bucket1",
                bucket_location="cn-north-1",
                kms_id="1f26f8d8-65d4-436b-bea2-bd0ac1984f71",
                is_support_trace_files_encryption=False
            ),
            CheckBucketRequest(
                bucket_name="bucket2",
                bucket_location="cn-north-2",
                kms_id="0c1b7d87-5186-411a-86ce-ed3b2ec848c9",
                is_support_trace_files_encryption=False
            )
        ]
        request.body = CheckObsBucketsRequestBody(
            buckets=listBucketsbody
        )
```

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

## Go

### 检查obs桶状态请求体示例

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CheckObsBucketsRequest{}
    kmsIdBuckets:= "1f26f8d8-65d4-436b-bea2-bd0ac1984f71"
    isSupportTraceFilesEncryptionBuckets:= false
    kmsIdBuckets1:= "0c1b7d87-5186-411a-86ce-ed3b2ec848c9"
    isSupportTraceFilesEncryptionBuckets1:= false
    var listBucketsbody = []model.CheckBucketRequest{
        {
            BucketName: "bucket1",
            BucketLocation: "cn-north-1",
            KmsId: &kmsIdBuckets,
            IsSupportTraceFilesEncryption: &isSupportTraceFilesEncryptionBuckets,
        },
        {
            BucketName: "bucket2",
            BucketLocation: "cn-north-2",
            KmsId: &kmsIdBuckets1,
            IsSupportTraceFilesEncryption: &isSupportTraceFilesEncryptionBuckets1,
        },
    }
    request.Body = &model.CheckObsBucketsRequestBody{
        Buckets: &listBucketsbody,
    }
    response, err := client.CheckObsBuckets(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

## 更多

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

## 状态码

状态码	描述
200	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求资源不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.4.5 查询事件的资源类型列表

### 功能介绍

查询事件的资源类型列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{domain\_id}/resources

表 6-155 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	账户id，参见 <a href="#">获取账号ID和项目ID</a> 章节。

## 请求参数

无

## 响应参数

状态码： 200

表 6-156 响应 Body 参数

参数	参数类型	描述
resources	Array of <a href="#">TraceResource</a> objects	返回的资源类型列表。

表 6-157 TraceResource

参数	参数类型	描述
service_type	String	云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。
resource	Array of strings	云服务对应的资源类型列表。

状态码： 400

表 6-158 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

状态码： 401

表 6-159 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识，CTS.XXX。
error_msg	String	错误描述。

状态码： 403

表 6-160 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 404

表 6-161 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 500

表 6-162 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 503

表 6-163 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

```
GET https://{endpoint}/v3/{domain_id}/resources
```

## 响应示例

状态码: 200

请求成功。

```
{  
  "resources": [ {
```



```
"service_type": "CTS",  
"resource": [ "trace", "notification", "tracker" ]  
}, {  
"service_type": "AOM",  
"resource": [ "xxx1", "xxx2" ]  
}]  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;  
import com.huaweicloud.sdk.cts.v3.*;  
import com.huaweicloud.sdk.cts.v3.model.*;  
  
public class ListTraceResourcesSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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 BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        CtsClient client = CtsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListTraceResourcesRequest request = new ListTraceResourcesRequest();  
        try {  
            ListTraceResourcesResponse response = client.listTraceResources(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

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

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

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

## Go

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

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

## 更多

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

## 状态码

状态码	描述
200	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求资源不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

# 6.5 标签管理

## 6.5.1 批量添加 CTS 资源标签

### 功能介绍

批量添加CTS资源标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/{resource\_type}/{resource\_id}/tags/create

表 6-164 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 参见 <a href="#">获取账号ID和项目ID</a> 章节。
resource_type	是	String	CTS服务的资源类型, 目前仅支持cts-tracker。 枚举值: <ul style="list-style-type: none"><li>• cts-tracker</li></ul>
resource_id	是	String	资源ID。

## 请求参数

表 6-165 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of <b>Tags</b> objects	标签列表。

表 6-166 Tags

参数	是否必选	参数类型	描述
key	否	String	键。最大长度128个unicode字符。标签的键可以包含任意语种字母、数字、空格和_.: = + - @, 但首尾不能含有空格, 不能以_sys_开头。
value	否	String	值。每个值最大长度255个unicode字符, 删除时如果value有值按照key/value删除, 如果value没值, 则按照key删除。标签的值可以包含任意语种字母、数字、空格和_./ = + - @, 但首尾不能含有空格。

## 响应参数

状态码: 401

表 6-167 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 403

表 6-168 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 404

表 6-169 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 500

表 6-170 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码: 503

表 6-171 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

追踪器标签创建示例。

```
POST https://{endpoint}/v3/{project_id}/{resource_type}/{resource_id}/tags/create
{
  "tags": [ {
    "key": "111",
    "value": "33"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

追踪器标签创建示例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

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

public class BatchCreateResourceTagsSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateResourceTagsRequest request = new BatchCreateResourceTagsRequest();
        BatchCreateResourceTagsRequestBody body = new BatchCreateResourceTagsRequestBody();
        List<Tags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tags()
                .withKey("111")
                .withValue("33")
        );
    }
}
```

```
body.withTags(listbodyTags);
request.withBody(body);
try {
    BatchCreateResourceTagsResponse response = client.batchCreateResourceTags(request);
    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

追踪器标签创建示例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = BatchCreateResourceTagsRequest()
        listTagsbody = [
            Tags(
                key="111",
                value="33"
            )
        ]
        request.body = BatchCreateResourceTagsRequestBody(
            tags=listTagsbody
        )
        response = client.batch_create_resource_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

追踪器标签创建示例。

```
package main

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

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

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

    client := cts.NewCtsClient(
        cts.CtsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateResourceTagsRequest{}
    keyTags := "111"
    valueTags := "33"
    var listTagsbody = []model.Tags{
        {
            Key: &keyTags,
            Value: &valueTags,
        },
    }
    request.Body = &model.BatchCreateResourceTagsRequestBody{
        Tags: &listTagsbody,
    }
    response, err := client.BatchCreateResourceTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

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

## 状态码

状态码	描述
200	创建成功。



状态码	描述
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求中的资源不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 6.5.2 批量删除 CTS 资源标签

### 功能介绍

批量删除CTS资源标签。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{project\_id}/{resource\_type}/{resource\_id}/tags/delete

表 6-172 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，参见 <a href="#">获取账号ID和项目ID</a> 章节。
resource_type	是	String	CTS服务的资源类型，目前仅支持cts-tracker。 枚举值： • <b>cts-tracker</b>
resource_id	是	String	资源ID。

## 请求参数

表 6-173 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of <b>Tags</b> objects	标签列表。

表 6-174 Tags

参数	是否必选	参数类型	描述
key	否	String	键。最大长度128个unicode字符。标签的键可以包含任意语种字母、数字、空格和_ . : = + - @, 但首尾不能含有空格, 不能以_sys_开头。
value	否	String	值。每个值最大长度255个unicode字符, 删除时如果value有值按照key/value删除, 如果value没值, 则按照key删除。标签的值可以包含任意语种字母、数字、空格和_ . : / = + - @, 但首尾不能含有空格。

## 响应参数

状态码： 401

表 6-175 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

状态码： 403

表 6-176 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

### 状态码：404

表 6-177 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

### 状态码：500

表 6-178 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

### 状态码：503

表 6-179 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码标识, CTS.XXX。
error_msg	String	错误描述。

## 请求示例

批量删除追踪器标签示例。

```
DELETE https://{endpoint}/v3/{project_id}/{resource_type}/{resource_id}/tags/delete
{
  "tags": [ {
    "key": "111",
    "value": "33"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

## Java

批量删除追踪器标签示例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cts.v3.region.CtsRegion;
import com.huaweicloud.sdk.cts.v3.*;
import com.huaweicloud.sdk.cts.v3.model.*;

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

public class BatchDeleteResourceTagsSolution {

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

        CtsClient client = CtsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CtsRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteResourceTagsRequest request = new BatchDeleteResourceTagsRequest();
        BatchDeleteResourceTagsRequestBody body = new BatchDeleteResourceTagsRequestBody();
        List<Tags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tags()
                .withKey("111")
                .withValue("33")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchDeleteResourceTagsResponse response = client.batchDeleteResourceTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

批量删除追踪器标签示例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcts.v3.region.cts_region import CtsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcts.v3 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = BatchDeleteResourceTagsRequest()
        listTagsbody = [
            Tags(
                key="111",
                value="33"
            )
        ]
        request.body = BatchDeleteResourceTagsRequestBody(
            tags=listTagsbody
        )
        response = client.batch_delete_resource_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

批量删除追踪器标签示例。

```
package main

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

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

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

```
Build()

client := cts.NewCtsClient(
    cts.CtsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchDeleteResourceTagsRequest{}
keyTags:= "111"
valueTags:= "33"
var listTagsbody = []model.Tags{
    {
        Key: &keyTags,
        Value: &valueTags,
    },
}
request.Body = &model.BatchDeleteResourceTagsRequestBody{
    Tags: &listTagsbody,
}
response, err := client.BatchDeleteResourceTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

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

## 状态码

状态码	描述
204	删除成功。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求中的资源不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

# 7 历史 API

## 7.1 API 版本号管理

### 7.1.1 查询 API 所有版本号

#### 功能介绍

查询云审计服务支持的所有API版本号。

#### URI

GET /

#### 请求参数

无

#### 响应参数

状态码： 200

表 7-1 响应 Body 参数

参数	参数类型	描述
versions	Array of <b>Versions</b> objects	描述versions相关对象的列表。

表 7-2 Versions

参数	参数类型	描述
id	String	版本ID（版本号），如v1。
links	Array of <a href="#">LinksBody</a> objects	API的URL地址。
version	String	若该版本API支持微版本，则填支持的最大微版本号，如果不支持微版本，则填空。
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。 枚举值： <ul style="list-style-type: none"><li>● CURRENT</li><li>● SUPPORTED</li><li>● DEPRECATED</li></ul>
updated	String	版本发布时间，要求用UTC时间表示。如v1发布的时间2014-06-28T12:20:21Z。
min_version	String	若该版本API支持微版本，则填支持的最小微版本号，如果不支持微版本，则填空。

表 7-3 LinksBody

参数	参数类型	描述
href	String	当前API版本的引用地址。
rel	String	当前API版本和被引用地址的关系。

## 请求示例

无

## 响应示例

状态码：200

请求正常。

```
{
  "versions": [
    {
      "id": "v3",
      "links": [
        {
          "href": "https://x.x.x.x/v3/",

```



```
    "rel": "self"
  }
],
"min_version": "",
"status": "CURRENT",
"updated": "2020-06-30T00:00:00Z",
"version": ""
},
{
  "id": "v2.0",
  "links": [
    {
      "href": "https://x.x.x.x/v2.0/",
      "rel": "self"
    }
  ],
  "min_version": "",
  "status": "DEPRECATED",
  "updated": "2018-09-30T00:00:00Z",
  "version": ""
},
{
  "id": "v1.0",
  "links": [
    {
      "href": "https://x.x.x.x/v1.0/",
      "rel": "self"
    }
  ],
  "min_version": "",
  "status": "DEPRECATED",
  "updated": "2018-09-30T00:00:00Z",
  "version": ""
}
]
```

## 状态码

状态码	描述
200	请求正常。
404	服务器无法找到被请求的资源。
500	请求未完成，服务异常。

## 错误码

请参见[错误码](#)。

## 7.1.2 查询 API 指定版本号

### 功能介绍

查询云审计服务指定API版本号。

### URI

GET /{version}

表 7-4 路径参数

参数	是否必选	参数类型	描述
version	是	String	版本号

## 请求参数

无

## 响应参数

状态码： 200

表 7-5 响应 Body 参数

参数	参数类型	描述
version	Object	描述version相关对象的列表。

表 7-6 Version

参数	参数类型	描述
id	String	版本ID（版本号），如v1。
links	Array of <a href="#">LinksBody</a> objects	API的URL地址。
version	String	若该版本API支持微版本，则填支持的最大微版本号，如果不支持微版本，则填空。
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。 枚举值： <ul style="list-style-type: none"><li>● CURRENT</li><li>● SUPPORTED</li><li>● DEPRECATED</li></ul>
updated	String	版本发布时间，要求用UTC时间表示。如v1发布的时间2014-06-28T12:20:21Z。
min_version	String	若该版本API支持微版本，则填支持的最小微版本号，如果不支持微版本，则填空。

表 7-7 LinksBody

参数	参数类型	描述
href	String	当前API版本的引用地址。
rel	String	当前API版本和被引用地址的关系。

## 请求示例

无

## 响应示例

**状态码： 200**

请求正常。

```
{
  "version": {
    "min_version": "",
    "links": [{
      "rel": "self",
      "href": "https://x.x.x.x/v3/"
    }
  ],
  "id": "v3",
  "updated": "2020-06-30T00:00:00Z",
  "version": "",
  "status": "CURRENT"
}
```

## 状态码

状态码	描述
200	请求正常。
404	服务器无法找到被请求的资源。
500	请求未完成，服务异常。

## 错误码

请参见[错误码](#)。

## 7.2 事件管理

## 7.2.1 查询事件列表(v1.0)

### 功能介绍

通过事件列表查询接口，可以查出系统记录的7天内资源操作记录。

### URI

GET /v1.0/{project\_id}/{tracker\_name}/trace

表 7-8 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见获取项目ID。
tracker_name	是	String	当前版本单租户仅支持一个追踪器，名称默认为“system”。

表 7-9 Query 参数

参数	是否必选	参数类型	描述
service_type	否	String	标识查询事件列表对应的云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以查看到该云服务的英文缩写。
user	否	String	标识特定用户名称，用以查询该用户下的所有事件。
from	否	Long	标识查询事件列表的起始时间戳（timestamp，为标准UTC时间，毫秒级，13位数字，不包括传入时间）默认为上一小时的时间戳。查询条件from与to配套使用。
limit	否	Integer	标示查询事件列表中限定返回的事件条数。不传时默认10条，最大值200条。

参数	是否必选	参数类型	描述
next	否	String	取值为响应中marker的值，用于标识查询事件的起始时间（自此条事件的记录时间起，向更早时间查询）。可以与“from”、“to”结合使用。最终的查询条件取两组时间条件的交集。
resource_id	否	String	标示查询事件列表对应的云服务资源ID。
resource_name	否	String	标示查询事件列表对应的资源名称。说明：该字段可能包含大写字母。
resource_type	否	String	标示查询事件列表对应的资源类型。
to	否	Long	标识查询事件列表的结束时间戳（timestamp，为标准UTC时间，毫秒级，13位数字，不包括传入时间）默认为当前时间戳。查询条件to与from配套使用。
trace_id	否	String	标示某一条事件的事件ID。当传入这个查询条件时，其他查询条件自动不生效。
trace_name	否	String	标示查询事件列表对应的事件名称。说明：该字段可能包含大写字母。
trace_rating	否	String	标示查询事件列表对应的事件等级目前有三种：正常(normal), 警告(warning), 事故(incident)。

## 请求参数

无

## 响应参数

状态码： 200

表 7-10 响应 Body 参数

参数	参数类型	描述
traces	Array of <a href="#">Traces</a> objects	本次查询事件列表返回事件数组。
meta_data	<a href="#">MetaData</a> object	本次查询事件条数和标记位。

表 7-11 Traces

参数	参数类型	描述
resource_id	String	标识事件对应的云服务资源ID。
trace_name	String	标识查询事件列表对应的事件名称。由0-9,a-z,A-Z,'-',',','_',组成，长度为1~64个字符，且以首字符必须为字母。
trace_rating	String	标识事件等级，目前有三种：正常（normal），警告（warning），事故（incident）。 枚举值： <ul style="list-style-type: none"><li>• normal</li><li>• warning</li><li>• incident</li></ul>
trace_type	String	标识事件发生源头类型，主要包括API调用（ApiCall），Console页面调用（ConsoleAction）和系统间调用（SystemAction）。
request	String	标识事件对应接口请求内容，即资源操作请求体。
response	String	记录用户请求的响应，标识事件对应接口响应内容，即资源操作结果返回体。
code	String	记录用户请求的响应，标识事件对应接口返回的HTTP状态码。
api_version	String	标识事件对应的云服务接口版本。
message	String	标识其他云服务为此条事件添加的备注信息。
record_time	Long	标识云审计服务记录本次事件的时间戳。
trace_id	String	标识事件的ID，由系统生成的UUID。
time	Long	标识事件产生的时间戳。
user	<a href="#">UserInfo</a> object	标识触发事件的用户信息。

参数	参数类型	描述
service_type	String	标识查询事件列表对应的云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以查看到该云服务的英文缩写。
resource_type	String	查询事件列表对应的资源类型。
source_ip	String	标识触发事件的租户IP。
resource_name	String	标识事件对应的资源名称。
request_id	String	记录本次请求的request id
location_info	String	记录本次请求出错后，问题定位所需要的辅助信息。
endpoint	String	云资源的详情页面
resource_url	String	云资源的详情页面的访问链接（不含endpoint）

表 7-12 UserInfo

参数	参数类型	描述
id	String	账号ID，获取方式请参见获取账号ID。
name	String	账号名称。
domain	<b>BaseUser</b> object	标识触发事件的用户domain信息。

表 7-13 BaseUser

参数	参数类型	描述
id	String	账号ID，获取方式请参见获取账号ID。
name	String	账号名称。

表 7-14 MetaData

参数	参数类型	描述
count	Integer	标识本次查询事件列表返回的事件记录的总条数。

参数	参数类型	描述
marker	String	标识本次查询事件列表返回的最后一个事件ID。可以使用这个参数返回值作为分页请求参数next的值，如果marker返回为null，则表示当前请求条件下查询事件列表已经全部返回没有下一页。

## 请求示例

```
GET https://{endpoint}/v1.0/{project_id}/{tracker_name}/trace?
limit=11&to=1479095278000&from=1478490478000&trace_name=createTracker&resource_type=tracker&service_type=CTS
```

## 响应示例

**状态码： 200**

请求正常。

```
{
  "meta_data" : {
    "count" : 2,
    "marker" : "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8"
  },
  "traces" : [ {
    "time" : 1472148708232,
    "user" : {
      "name" : "xxx",
      "domain" : {
        "name" : "xxx",
        "id" : "ded649d814464428ba89d04d7955c93e"
      }
    }
  },
  "response" : {
    "code" : "VPC.0514",
    "message" : "Update port fail."
  },
  "code" : 200,
  "service_type" : "VPC",
  "resource_type" : "eip",
  "resource_name" : "192.144.163.1",
  "resource_id" : "d502809d-0d1d-41ce-9690-784282142ccc",
  "trace_name" : "deleteEip",
  "trace_rating" : "warning",
  "trace_type" : "ConsoleAction",
  "api_version" : "2.0",
  "record_time" : 1481066128032,
  "trace_id" : "e001ccb9-bc09-11e6-b00b-4b2a61338db6"
}, {
  "time" : 1472148708232,
  "user" : {
    "name" : "xxx",
    "domain" : {
      "name" : "xxx",
      "id" : "ded649d814464428ba89d04d7955c93e"
    }
  },
  "response" : {
    "code" : "VPC.0514",
    "message" : "Update port fail."
  },
  "code" : 200,
  "service_type" : "VPC",
  "resource_type" : "eip",
```



```
"resource_name": "192.144.163.1",
"resource_id": "d502809d-0d1d-41ce-9690-784282142ccc",
"trace_name": "deleteEip",
"trace_rating": "warning",
"trace_type": "ConsoleAction",
"api_version": "2.0",
"record_time": 1481066128032,
"trace_id": "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8"
}]
}
```

## 状态码

状态码	描述
200	请求正常。
400	查询参数异常，请求未完成。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	查询事件不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 7.2.2 查询事件列表(v2.0)

### 功能介绍

通过事件列表查询接口，可以查出系统记录的7天内资源操作记录。

### URI

GET /v2.0/{project\_id}/{tracker\_name}/trace

表 7-15 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见获取项目ID。
tracker_name	是	String	当前版本单租户仅支持一个追踪器，名称默认为“system”。

表 7-16 Query 参数

参数	是否必选	参数类型	描述
service_type	否	String	标识查询事件列表对应的云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以查看到该云服务的英文缩写。
user	否	String	标识特定用户名称，用以查询该用户下的所有事件。
from	否	Long	标识查询事件列表的起始时间戳（timestamp，为标准UTC时间，毫秒级，13位数字，不包括传入时间）默认为上一小时的时间戳。查询条件from与to配套使用。
limit	否	Integer	标示查询事件列表中限定返回的事件条数。不传时默认10条，最大值200条。
next	否	String	取值为响应中marker的值，用于标识查询事件的起始时间（自此条事件的记录时间起，向更早时间查询）。可以与“from”、“to”结合使用。最终的查询条件取两组时间条件的交集。
resource_id	否	String	标示查询事件列表对应的云服务资源ID。
resource_name	否	String	标示查询事件列表对应的资源名称。说明：该字段可能包含大写字母。
resource_type	否	String	标示查询事件列表对应的资源类型。
to	否	Long	标识查询事件列表的结束时间戳（timestamp，为标准UTC时间，毫秒级，13位数字，不包括传入时间）默认为当前时间戳。查询条件to与from配套使用。
trace_id	否	String	标示某一条事件的事件ID。当传入这个查询条件时，其他查询条件自动不生效。

参数	是否必选	参数类型	描述
trace_name	否	String	标示查询事件列表对应的事件名称。说明：该字段可能包含大写字母。
trace_rating	否	String	标示查询事件列表对应的事件等级目前有三种：正常(normal),警告(warning),事故(incident)。

## 请求参数

无

## 响应参数

状态码： 200

表 7-17 响应 Body 参数

参数	参数类型	描述
traces	Array of <a href="#">Traces</a> objects	本次查询事件列表返回事件数组。
meta_data	<a href="#">MetaData</a> object	本次查询事件条数和标记位。

表 7-18 Traces

参数	参数类型	描述
resource_id	String	标识事件对应的云服务资源ID。
trace_name	String	标识查询事件列表对应的事件名称。由0-9,a-z,A-Z, '-', '_', ':' 组成，长度为1~64个字符，且以首字符必须为字母。
trace_rating	String	标识事件等级，目前有三种：正常（normal），警告（warning），事故（incident）。 枚举值： <ul style="list-style-type: none"><li>• normal</li><li>• warning</li><li>• incident</li></ul>

参数	参数类型	描述
trace_type	String	标识事件发生源头类型，主要包括API调用（ApiCall），Console页面调用（ConsoleAction）和系统间调用（SystemAction）。
request	String	标识事件对应接口请求内容，即资源操作请求体。
response	String	记录用户请求的响应，标识事件对应接口响应内容，即资源操作结果返回体。
code	String	记录用户请求的响应，标识事件对应接口返回的HTTP状态码。
api_version	String	标识事件对应的云服务接口版本。
message	String	标识其他云服务为此条事件添加的备注信息。
record_time	Long	标识云审计服务记录本次事件的时间戳。
trace_id	String	标识事件的ID，由系统生成的UUID。
time	Long	标识事件产生的时间戳。
user	UserInfo object	标识触发事件的用户信息。
service_type	String	标识查询事件列表对应的云服务类型。必须为已对接CTS的云服务的英文缩写，且服务类型一般为大写字母。已对接的云服务列表参见《云审计服务用户指南》“支持审计的服务及详细操作列表”章节，单击对应云服务的文档链接，可以看到该云服务的英文缩写。
resource_type	String	查询事件列表对应的资源类型。
source_ip	String	标识触发事件的租户IP。
resource_name	String	标识事件对应的资源名称。
request_id	String	记录本次请求的request id
location_info	String	记录本次请求出错后，问题定位所需要的辅助信息。
endpoint	String	云资源的详情页面
resource_url	String	云资源的详情页面的访问链接（不含endpoint）

表 7-19 UserInfo

参数	参数类型	描述
id	String	账号ID，获取方式请参见获取账号ID。
name	String	账号名称。
domain	BaseUser object	标识触发事件的用户domain信息。

表 7-20 BaseUser

参数	参数类型	描述
id	String	账号ID，获取方式请参见获取账号ID。
name	String	账号名称。

表 7-21 MetaData

参数	参数类型	描述
count	Integer	标识本次查询事件列表返回的事件记录的总条数。
marker	String	标识本次查询事件列表返回的最后一个事件ID。可以使用这个参数返回值作为分页请求参数next的值，如果marker返回为null，则表示当前请求条件下查询事件列表已经全部返回没有下一页。

## 请求示例

```
GET https://{endpoint}/v2.0/{project_id}/{tracker_name}/trace?
limit=11&to=1479095278000&from=1478490478000&trace_name=createTracker&resource_type=tracker&service_type=CTS
```

## 响应示例

**状态码： 200**

请求正常。

```
{
  "meta_data": {
    "count": 2,
    "marker": "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8"
  },
  "traces": [ {
    "time": 1472148708232,
    "user": {
      "name": "xxx",
      "domain": {
        "name": "xxx",
        "id": "ded649d814464428ba89d04d7955c93e"
      }
    }
  }
]
```

```
}
},
"response": {
  "code": "VPC.0514",
  "message": "Update port fail."
},
"code": 200,
"service_type": "VPC",
"resource_type": "eip",
"resource_name": "192.144.163.1",
"resource_id": "d502809d-0d1d-41ce-9690-784282142ccc",
"trace_name": "deleteEip",
"trace_rating": "warning",
"trace_type": "ConsoleAction",
"api_version": "2.0",
"record_time": 1481066128032,
"trace_id": "e001ccb9-bc09-11e6-b00b-4b2a61338db6"
}, {
  "time": 1472148708232,
  "user": {
    "name": "xxx",
    "domain": {
      "name": "xxx",
      "id": "ded649d814464428ba89d04d7955c93e"
    }
  },
  "response": {
    "code": "VPC.0514",
    "message": "Update port fail."
  },
  "code": 200,
  "service_type": "VPC",
  "resource_type": "eip",
  "resource_name": "192.144.163.1",
  "resource_id": "d502809d-0d1d-41ce-9690-784282142ccc",
  "trace_name": "deleteEip",
  "trace_rating": "warning",
  "trace_type": "ConsoleAction",
  "api_version": "2.0",
  "record_time": 1481066128032,
  "trace_id": "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8"
}
}]
}
```

## 状态码

状态码	描述
200	请求正常。
400	查询参数异常，请求未完成。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	查询事件不存在，请求未完成。
500	服务内部异常，请求未完成。
503	被请求的服务无效。建议直接修改该请求，不要重试该请求。

## 错误码

请参见[错误码](#)。

## 7.3 追踪器管理

### 7.3.1 创建追踪器

#### 功能介绍

云审计服务开通后系统会自动创建一个追踪器，用来关联系统记录的所有操作。目前，一个云账户在一个Region下仅支持创建一个管理类追踪器。云审计服务支持在管理控制台查询近7天内的操作记录。如需保存更长时间的操作记录，您可以在创建追踪器之后通过对象存储服务（Object Storage Service，以下简称OBS）将操作记录实时保存至OBS桶中。

#### URI

POST /v1.0/{project\_id}/tracker

表 7-22 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见获取项目ID。

#### 请求参数

表 7-23 请求 Body 参数

参数	是否必选	参数类型	描述
bucket_name	是	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。
file_prefix_name	否	String	标识需要存储于OBS的日志文件前缀，0-9，a-z，A-Z，'-', '!', '_'长度为0~64字符。

参数	是否必选	参数类型	描述
is_obs_created	否	Boolean	是否支持新建OBS桶,默认为“false”。值为“true”时,表示新建OBS桶存储事件文件; 值为“false”时,选择已存在的OBS桶存储事件文件。桶名称包含数字、字母、'-'和'.'(非数字、字母类字符最多存在1个), 长度为3~64字符。
is_support_trace_files_encryption	否	Boolean	事件文件转储加密功能开关。该参数必须与kms_id参数同时使用。
kms_id	否	String	事件文件转储加密所采用的密钥id(从KMS获取)。当"is_support_trace_files_encryption"参数值为“是”时,此参数为必选项。
lts	否	Lts object	事件分析
log_file_validate	否	LogFileValidate object	文件校验

表 7-24 Lts

参数	是否必选	参数类型	描述
is_lts_enabled	是	Boolean	是否启用日志服务检索功能。
log_group_name	是	String	云审计服务在日志服务中创建的日志组名称。
log_topic_name	是	String	云审计服务在日志服务中创建的日志主题名称。

表 7-25 LogFileValidate

参数	是否必选	参数类型	描述
is_support_validate	是	Boolean	是否打开事件文件校验。

## 响应参数

状态码： 201



表 7-26 响应 Body 参数

参数	参数类型	描述
id	String	追踪器唯一标识。
create_time	Long	追踪器创建时间戳。
domain_id	String	账号ID, 获取方式请参见获取账号ID。
project_id	String	项目ID。
is_support_trace_files_encryption	Boolean	事件文件转储加密功能开关。该参数必须与 kms_id 参数同时使用。
kms_id	String	事件文件转储加密所采用的密钥id (从KMS获取)。当"is_support_trace_files_encryption"参数值为“是”时, 此参数为必选项。
obs_info	<b>ObsInfo</b> object	转储桶信息。
status	String	标识追踪器状态, 该接口返回正常状态 (enabled)。
tracker_name	String	追踪器名称。
tracker_type	String	追踪器类型。
group_id	String	日志组唯一标识。
stream_id	String	日志流唯一标识。
lts	<b>Lts</b> object	事件分析
is_support_validate	Boolean	是否打开事件文件校验。当前环境仅 "tracker_type" 参数值为 "system" 时支持该功能。

表 7-27 ObsInfo

参数	参数类型	描述
bucket_name	String	标识OBS桶名称。由数字或字母开头, 支持小写字母、数字、“-”、“.”, 长度为3~63个字符。
file_prefix_name	String	标识需要存储于OBS的日志文件前缀, 0-9, a-z, A-Z, '-', '!', '_' 长度为0~64字符。
is_obs_created	Boolean	标识配置桶是否由追踪器自动创建。
is_authorized_bucket	Boolean	标识配置桶是否已经授权给CTS服务账号。
bucket_lifecycle	String	标识配置桶内对象存储周期。当"tracker_type" 参数值为"data"时该参数值有效。

表 7-28 Lts

参数	参数类型	描述
is_lts_enabled	Boolean	是否启用日志服务检索功能。
log_group_name	String	云审计服务在日志服务中创建的日志组名称。
log_topic_name	String	云审计服务在日志服务中创建的日志主题名称。

## 请求示例

```
POST https://{endpoint}/v1.0/{project_id}/tracker
{
  "bucket_name": "obs-f1da",
  "is_support_trace_files_encryption": true,
  "kms_id": "13a4207c-7abe-4b68-8510-16b84c3b5504",
  "is_obs_created": true,
  "file_prefix_name": "yO8Q",
  "lts": {
    "is_lts_enabled": true,
    "log_group_name": "CTS",
    "log_topic_name": "system-trace"
  },
  "log_file_validate": {
    "is_support_validate": true
  }
}
```

## 响应示例

状态码： 201

请求成功。

```
{
  "id": "2e6fa9b8-8c6e-456d-b5d3-77be972d220b",
  "create_time": 1587958482923,
  "domain_id": "aexxxxxxxxx4d4fb4bexxxxxx791fbf",
  "is_support_trace_files_encryption": true,
  "kms_id": "13a4207c-7abe-4b68-8510-16b84c3b5504",
  "obs_info": {
    "bucket_name": "obs-f1da",
    "file_prefix_name": "yO8Q"
  },
  "project_id": "bb1xxxxxxxxe4f498cbxxxxxxx35634",
  "lts": {
    "is_lts_enabled": true,
    "log_group_name": "CTS",
    "log_topic_name": "system-trace"
  },
  "log_file_validate": {
    "is_support_validate": true
  },
  "tracker_name": "system",
  "tracker_type": "system",
  "status": "enabled"
}
```

## 状态码

状态码	描述
201	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	请求中的OBS桶不存在，请求未完成。
500	服务内部异常，请求未完成。

## 错误码

请参见[错误码](#)。

### 7.3.2 查询追踪器

#### 功能介绍

开通云审计服务成功后，您可以在追踪器信息页面查看系统自动创建的追踪器的详细信息。详细信息主要包括追踪器名称，用于存储操作事件的OBS桶名称和OBS桶中的事件文件前缀。

#### URI

GET /v1.0/{project\_id}/tracker

表 7-29 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见获取项目ID。

表 7-30 Query 参数

参数	是否必选	参数类型	描述
tracker_name	是	String	标示追踪器名称。在不传入该字段的情况下，将查询租户所有的追踪器。当前版本单租户仅支持一个追踪器，名称默认为“system”。

## 请求参数

无

## 响应参数

状态码： 200

表 7-31 响应 Body 参数

参数	参数类型	描述
id	String	追踪器唯一标识。
create_time	Long	追踪器创建时间戳。
domain_id	String	账号ID，获取方式请参见获取账号ID。
project_id	String	项目ID。
tracker_name	String	标识追踪器名称，当前版本默认为“system”。
tracker_type	String	标识追踪器类型。
bucket_name	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。
file_prefix_name	String	标识需要存储于OBS的日志文件前缀。
status	String	标识追踪器状态，包括正常（enabled），停止（disabled）和异常（error）三种状态，状态为异常时需通过明细（detail）字段说明错误来源。 枚举值： <ul style="list-style-type: none"><li>• enabled</li><li>• disabled</li></ul>
detail	String	该参数仅在追踪器状态异常时返回，用于标识追踪器异常的原因，包括桶策略异常（bucketPolicyError），桶不存在（noBucket）和欠费或冻结（arrears）三种原因。
is_obs_created	Boolean	是否支持新建OBS桶。值为“true”时，表示新建OBS桶存储事件文件；值为“false”时，选择已存在的OBS桶存储事件文件。桶名称包含数字、字母、'-'和'.'（非数字、字母类字符最多存在1个），长度为3~64字符。
is_support_trace_files_encryption	Boolean	事件文件转储加密功能开关。该参数必须与kms_id参数同时使用。
kms_id	String	事件文件转储加密所采用的密钥id（从KMS获取）。当“is_support_trace_files_encryption”参数值为“是”时，此参数为必选项。

参数	参数类型	描述
group_id	String	日志组唯一标识。
stream_id	String	日志流唯一标识。
lts	Lts object	事件分析
is_support_validate	Boolean	是否打开事件文件校验。

表 7-32 Lts

参数	参数类型	描述
is_lts_enabled	Boolean	是否启用日志服务检索功能。
log_group_name	String	云审计服务在日志服务中创建的日志组名称。
log_topic_name	String	云审计服务在日志服务中创建的日志主题名称。

## 请求示例

```
GET https://{endpoint}/v1.0/{project_id}/tracker?tracker_name=system
```

## 响应示例

状态码： 200

请求成功。

```
{
  "id": "2e6fa9b8-8c6e-456d-b5d3-77be972d220b",
  "create_time": 1587958482923,
  "domain_id": "aexxxxxxx4d4fb4bexxxxxx791fbf",
  "is_support_trace_files_encryption": true,
  "kms_id": "13a4207c-7abe-4b68-8510-16b84c3b5504",
  "bucket_name": "obs-f1da",
  "file_prefix_name": "yO8Q",
  "project_id": "bb1xxxxxxe4f498cbxxxxxxx35634",
  "lts": {
    "is_lts_enabled": true,
    "log_group_name": "CTS",
    "log_topic_name": "system-trace"
  },
  "log_file_validate": {
    "is_support_validate": true
  },
  "tracker_name": "system",
  "tracker_type": "system",
  "status": "enabled"
}
```

## 状态码

状态码	描述
200	请求成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
500	服务内部异常，请求未完成。

## 错误码

请参见[错误码](#)。

### 7.3.3 删除追踪器

#### 功能介绍

云审计服务支持删除已创建的追踪器。删除追踪器对已有的操作记录没有影响，当您重新开通云审计服务后，依旧可以查看已有的操作记录。

#### URI

DELETE /v1.0/{project\_id}/tracker

表 7-33 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见获取项目ID。

表 7-34 Query 参数

参数	是否必选	参数类型	描述
tracker_name	是	String	标识追踪器名称。在不传入该字段的情况下，将删除当前租户所有的追踪器。当前版本单租户仅支持一个追踪器，名称默认为“system”。

#### 请求参数

无

## 响应参数

无

## 请求示例

```
DELETE https://{endpoint}/v1.0/{project_id}/tracker?tracker_name=system
```

## 响应示例

无

## 状态码

状态码	描述
204	删除成功。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	服务器无法找到被请求的资源或部分追踪器删除失败。
500	服务内部异常，请求未完成；或部分追踪器删除失败。

## 错误码

请参见[错误码](#)。

## 7.3.4 修改追踪器

### 功能介绍

云审计服务支持修改已创建追踪器的配置项，包括OBS桶转储、关键事件通知、事件转储加密、通过LTS对管理类事件进行检索、事件文件完整性校验以及追踪器启停状态等相关参数，修改追踪器对已有的操作记录没有影响。修改追踪器完成后，系统立即以新的规则开始记录操作。

### URI

```
PUT /v1.0/{project_id}/tracker/{tracker_name}
```

表 7-35 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见获取项目ID。

参数	是否必选	参数类型	描述
tracker_name	是	String	标识追踪器名称，当前版本单租户仅支持一个追踪器，名称默认为“system”。

## 请求参数

表 7-36 请求 Body 参数

参数	是否必选	参数类型	描述
bucket_name	否	String	标识OBS桶名称。由数字或字母开头，支持小写字母、数字、“-”、“.”，长度为3~63个字符。
file_prefix_name	否	String	标识需要存储于OBS的日志文件前缀，0-9, a-z, A-Z, '-', '!', '_'长度为0~64字符。
status	否	String	标识追踪器状态，该接口中可修改的状态包括正常（enabled）和停止（disabled）。如果选择修改状态为停止，则修改成功后追踪器停止记录事件。 枚举值： <ul style="list-style-type: none"><li>• enabled</li><li>• disabled</li></ul>
is_obs_created	否	Boolean	是否支持新建OBS桶。值为“true”时，表示新建OBS桶存储事件文件；值为“false”时，选择已存在的OBS桶存储事件文件。桶名称包含数字、字母、'-'和'!'（非数字、字母类字符最多存在1个），长度为3~64字符。
is_support_trace_files_encryption	否	Boolean	事件文件转储加密功能开关。该参数必须与kms_id参数同时使用。
kms_id	否	String	事件文件转储加密所采用的密钥id（从KMS获取）。当"is_support_trace_files_encryption"参数值为“是”时，此参数为必选项。
lts	否	Lts object	事件分析



参数	是否必选	参数类型	描述
log_file_validate	否	LogFileValidate object	文件校验

表 7-37 Lts

参数	是否必选	参数类型	描述
is_lts_enabled	是	Boolean	是否启用日志服务检索功能。
log_group_name	是	String	云审计服务在日志服务中创建的日志组名称。
log_topic_name	是	String	云审计服务在日志服务中创建的日志主题名称。

表 7-38 LogFileValidate

参数	是否必选	参数类型	描述
is_support_validate	是	Boolean	是否打开事件文件校验。

## 响应参数

无

## 请求示例

```
PUT https://{endpoint}/v1.0/{project_id}/tracker/system
```

```
{
  "bucket_name": "my_created_bucket",
  "file_prefix_name": "some_folder",
  "is_obs_created": true,
  "is_support_trace_files_encryption": true,
  "kms_id": "13a4207c-7abe-4b68-8510-16b84c3b5504",
  "lts": {
    "is_lts_enabled": true,
    "log_group_name": "CTS",
    "log_topic_name": 'system-trace'
  },
  "log_file_validate": {
    "is_support_validate": true
  },
  "status": "disabled"
}
```

## 响应示例

无

## 状态码

状态码	描述
200	请求正常。
400	服务器未能处理请求。
401	请求鉴权校验失败，访问被拒绝。
403	请求权限校验失败，访问被禁止。
404	服务器无法找到被请求的资源。
500	服务内部异常，请求未完成；或部分追踪器删除失败。

## 错误码

请参见[错误码](#)。

# 8 权限和授权项

如果您需要对您所拥有的CTS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CTS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或角色，才能使用户组中的用户获得对应的权限，这一过程称为授权。授权后，用户就可以基于被授予的权限对CTS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最初提供了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事件，那么这个IAM用户被授予的策略中必须包含允许“cts:trace:list”的授权项，该接口才能调用成功。

## 支持的授权项

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

- 权限：自定义策略中授权项定义的内容即为权限。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。
- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。
- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项

对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。关于IAM项目与企业项目的区别，详情请参见[IAM与企业管理的区别](#)。

### 说明

“√”表示支持，“x”表示暂不支持。

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询事件列表	GET /v3/{project_id}/traces	cts:trace:list	-	√	x
查询事件列表	GET /v2.0/{project_id}/{tracker_name}/trace	cts:trace:list	-	√	x
查询事件列表	GET /v1.0/{project_id}/{tracker_name}/trace	cts:trace:list	-	√	x
查询追踪器	GET /v3/{project_id}/trackers	cts:tracker:list	obs:bucket:GetBucketAcl obs:bucket:ListAllMyBuckets	√	x
查询追踪器	GET /v1.0/{project_id}/tracker	cts:tracker:list	obs:bucket:GetBucketAcl obs:bucket:ListAllMyBuckets	√	x

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM 项目 (Project)	企业项目 (Enterprise Project)
创建追踪器	POST /v3/{project_id}/tracker	cts:tracker:create	lts:topics:list lts:topics:create lts:groups:list lts:groups:create obs:bucket:CreateBucket obs:bucket:HeadBucket obs:bucket:GetLifecycleConfiguration obs:bucket:PutLifecycleConfiguration obs:bucket:GetBucketAcl obs:bucket:PutBucketAcl obs:bucket:PutBucketAclkms:cmk:list	√	x
创建追踪器	POST /v1.0/{project_id}/tracker	cts:tracker:create	lts:topics:list lts:topics:create lts:groups:list lts:groups:create obs:bucket:CreateBucket obs:bucket:HeadBucket obs:bucket:GetLifecycleConfiguration obs:bucket:PutLifecycleConfiguration obs:bucket:GetBucketAcl obs:bucket:PutBucketAcl obs:bucket:PutBucketAclkms:cmk:list	√	x

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM 项目 (Project)	企业项目 (Enterprise Project)
修改追踪器	PUT /v3/{project_id}/tracker	cts:tracker:update	lts:topics:list lts:topics:create lts:groups:list lts:groups:create obs:bucket:CreateBucket obs:bucket:HeadBucket obs:bucket:GetLifecycleConfiguration obs:bucket:PutLifecycleConfiguration obs:bucket:GetBucketAcl obs:bucket:PutBucketAcl kms:cmk:list	√	x
修改追踪器	PUT /v1.0/{project_id}/tracker/{tracker_name}	cts:tracker:update	lts:topics:list lts:topics:create lts:groups:list lts:groups:create obs:bucket:CreateBucket obs:bucket:HeadBucket obs:bucket:GetLifecycleConfiguration obs:bucket:PutLifecycleConfiguration obs:bucket:GetBucketAcl obs:bucket:PutBucketAcl kms:cmk:list	√	x
删除追踪器	DELETE /v3/{project_id}/trackers	cts:tracker:delete	-	√	x

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM 项目 (Project)	企业项目 (Enterprise Project)
删除追踪器	DELETE /v1.0/{project_id}/tracker	cts:tracker:delete	-	√	x
创建关键操作通知	POST /v3/{project_id}/notifications	cts:notification:create	smn:topic:listTopic iam:users:listUsers iam:groups:listGroups	√	x
修改关键操作通知	PUT /v3/{project_id}/notifications	cts:notification:update	smn:topic:listTopic iam:users:listUsers iam:groups:listGroups	√	x
删除关键操作通知	DELETE /v3/{project_id}/notifications	cts:notification:delete	-	√	x
查询关键操作通知	GET /v3/{project_id}/notifications/{notification_type}	cts:notification:list	-	√	x
查询租户追踪器配额信息	GET /v3/{project_id}/quotas	cts:quota:get	-	√	x
查询云服务的全量操作列表	GET /v3/{project_id}/operations	cts:notification:listOperation	-	√	x
查询审计日志的操作用户列表	GET /v3/{project_id}/user-resources	cts:tracker:listTraceUser	-	√	x

权限	对应API接口	授权项 (Action)	依赖的授权项	IAM 项目 (Project)	企业项目 (Enterprise Project)
检查已经配置 OBS 桶是否可以成功转储	POST /v3/{domain_id}/checkbucket	cts:tracker:list	obs:bucket:ListAllMyBuckets	√	x
查询事件的资源类型列表	GET /v3/{domain_id}/resources	cts:trace:listResource	-	√	x
批量添加 CTS 资源标签	POST /v3/{project_id}/{resource_type}/{resource_id}/tags/create	cts:tag:create	-	√	x
批量删除 CTS 资源标签	DELETE /v3/{project_id}/{resource_type}/{resource_id}/tags/delete	cts:tag:delete	-	√	x



# 9 附录

## 9.1 错误码

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。

更多服务错误码请参见[API错误中心](#)。

状态码	错误码	错误信息	描述	处理措施
400	CTS.0001	The IAM or OBS service is abnormal.	IAM或OBS服务异常	请联系技术支持
400	CTS.0003	The message body is empty or invalid.	body体为空或非法	请校验body内容和格式
400	CTS.0200	The number of trackers has reached the upper limit.	追踪器数量已满	请删除或修改不需要的追踪器
400	CTS.0201	A management tracker has been created.	已有管理类追踪器	请检查是否已生效
400	CTS.0202	The value of the tracker_type parameter is incorrect.	tracker_type 字段不符合格式	请将对应值改为 system或data

状态码	错误码	错误信息	描述	处理措施
400	CTS.0203	The value of tracker_name parameter is in an incorrect format.	tracker_name 字段不符合格式	请参考参数描述进行修改
400	CTS.0204	The tracker_name parameter of a management tracker can only be set to system.	管理类追踪器 tracker_name 字段应为 system	请参考参数描述进行修改
400	CTS.0205	The status parameter can only be set to enabled or disabled.	status 字段只能为 enabled 或 disabled	请将对应值改为 enabled 或 disabled
400	CTS.0206	The data_bucket parameter cannot be included in the message body for a management tracker.	管理类追踪器 body 不能有 data_bucket 参数	请去掉 data_bucket 参数
400	CTS.0207	The tracker_name parameter in the message body cannot be set to system for a data tracker.	数据类追踪器 body  tracker_name 不能为 system	请将 tracker_name 改为除 system 以外的值
400	CTS.0208	The tracker already exists.	已存在该追踪器	请检查该追踪器是否已经存在
400	CTS.0209	A type of operations on an OBS bucket can be tracked by only one tracker.	追踪一个桶的同一个操作类型	请更改追踪项

状态码	错误码	错误信息	描述	处理措施
400	CTS.0210	The OBS bucket to track cannot be empty.	追踪的桶不能为空	请换一个桶或使桶不为空
400	CTS.0211	The tracked OBS bucket does not exist.	被追踪的桶不存在	请检查 bucket_name 是否正确填写
400	CTS.0212	The tracked OBS bucket cannot be modified.	被追踪的obs桶不可修改	请撤回桶的更改
400	CTS.0213	The OBS bucket used for trace transfer cannot be a tracked OBS bucket.	被追踪的桶与转储的桶相同	请更换转储的桶
400	CTS.0215	The OBS bucket already exists.	桶已经存在	请修改 bucket_name
400	CTS.0216	Failed to create a bucket.	创建桶失败	请联系技术支持
400	CTS.0217	Failed to set a lifecycle rule for the OBS bucket.	设置桶生命周期规则失败	请联系技术支持
400	CTS.0218	The value of file_prefix_name is in an incorrect format.	file_prefix_name 字段不符合格式	请参考参数描述进行修改
400	CTS.0219	The operation type cannot be empty.	操作类型不能为空	请选择至少一个追踪操作
400	CTS.0220	KMS is not supported.	不支持KMS	请联系技术支持
400	CTS.0221	The KMS ID is empty.	KMS_ID 为空	请检查 kms_id 是否正确

状态码	错误码	错误信息	描述	处理措施
400	CTS.0222	KMS verification failed.	KMS校验失败	请检查kms_id是否正确
400	CTS.0225	Only WRITE and/or READ operations on the OBS bucket can be tracked.	追踪桶操作类型必须为WRITE、READ或WRITE和READ。	检查参数是否传入正确。
400	CTS.0228	The CTS service is not trusted.	云审计服务不受信任。	请在org服务界面开启可信服务。
400	CTS.0229	The organization tracker already exists.	组织追踪器已存在。	请先关闭已开启的组织追踪器。
400	CTS.0231	Invalid bucket name. A bucket name must be a string of 3 to 63 characters, including only lowercase letters, digits, hyphens (-), or periods (.). It must start with a digit or a lowercase letter.	无效的桶名称。bucket_name是3到63个字符的字符串，以数字或字母开头，支持小写字母，数字，“-”或“.”。	检查桶名称填写是否正确。
400	CTS.0300	Query failed.	query查询失败	请稍后重试或联系技术支持
403	CTS.0002	Authentication failed or you do not have the permissions required.	用户鉴权失败或没有权限	请检查用户权限
403	CTS.0013	No permission, Please check roles.	无对应操作权限	请配置对应操作权限

状态码	错误码	错误信息	描述	处理措施
404	CTS.0100	API version query is not supported in CTS.	CTS不支持查询接口版本号	请联系技术支持
404	CTS.0214	The tracker does not exist.	追踪器不存在	请检查该追踪器是否已删除
500	CTS.0004	Failed to write data.	写入数据失败	请联系技术支持
500	CTS.0005	Failed to read data.	读取数据失败	请联系技术支持

## 9.2 获取账号 ID 和项目 ID

### 从控制台获取账号 ID 和项目 ID

在调用接口的时候，部分URL中需要填入账号ID（domain-id）和项目ID，您可以通过控制台获取这些参数，步骤如下：

1. 注册并登录管理控制台。单击用户名，在下拉列表中单击“我的凭证”。



2. 在“我的凭证”页面查看账号ID和项目ID。



多项目时，展开“所属区域”，从“项目ID”列获取子项目ID。

## 调用 API 获取项目 ID

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点。

响应示例如下，其中projects下的“id”即为项目ID。

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180xxxx",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d1xxxx",
      "name": "xx-region-1",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f89xxxx"
      },
      "id": "a4a5d4098fb4474fa22cd0xxxx",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

# A 修订记录

发布日期	修订记录
2024-03-28	<p>第二十四次正式发布。</p> <p>本次更新说明如下：</p> <ul style="list-style-type: none"><li>• 创建关键操作通知：新增agency_name字段。</li><li>• 修改关键操作通知：新增agency_name字段。</li><li>• 查询事件列表：新增 access_key_id、enterprise_project_id、resource_account_id和user等字段。</li><li>• 创建追踪器：新增agency_name字段。</li><li>• 修改追踪器：新增agency_name字段。</li><li>• 查询追踪器：新增agency_name字段。</li></ul>
2023-9-21	<p>第二十三次正式发布。</p> <p>本次变更说明如下：</p> <p>追踪器管理接口新增is_organization_tracker、management_event_selector参数说明。</p>
2023-09-12	<p>第二十二次正式发布。</p> <p>本次变更说明如下：</p> <ul style="list-style-type: none"><li>• 新增以下接口（V3）：<ul style="list-style-type: none"><li>- 查询云服务的全量操作列表</li><li>- 查询审计日志的操作用户列表</li><li>- 检查已经配置OBS桶是否可以成功转储</li><li>- 查询事件的资源类型列表</li><li>- 批量添加CTS资源标签</li><li>- 批量删除CTS资源标签</li></ul></li></ul>
2023-04-20	<p>第二十一次正式发布。</p> <p>本次变更说明如下：</p> <p>优化API V3 “创建追踪器” 的响应示例。</p>

发布日期	修订记录
2021-10-15	<p>第二十次正式发布。</p> <p>本次变更说明如下：</p> <ul style="list-style-type: none"> <li>● 新增以下接口（V3）： <ul style="list-style-type: none"> <li>- 创建关键操作通知</li> <li>- 修改关键操作通知</li> <li>- 查询关键操作通知</li> <li>- 删除关键操作通知</li> </ul> </li> </ul>
2020-08-18	<p>第十九次正式发布。</p> <p>本次变更说明如下：</p> <ul style="list-style-type: none"> <li>● “API（V1）”更名为“历史API”</li> <li>● 新增以下章节 <ul style="list-style-type: none"> <li>- 调用说明</li> <li>- 终端节点</li> <li>- 约束与限制</li> <li>- 基本概念</li> <li>- API版本选择建议</li> </ul> </li> </ul>
2020-06-30	<p>第十八次正式发布。</p> <p>本次变更说明如下：</p> <ul style="list-style-type: none"> <li>● 新增以下接口（V3）： <ul style="list-style-type: none"> <li>- 查询事件列表</li> <li>- 创建追踪器</li> <li>- 修改追踪器</li> <li>- 查询追踪器</li> <li>- 删除追踪器</li> <li>- 查询租户追踪器配额信息</li> </ul> </li> <li>● 新增以下应用示例： <ul style="list-style-type: none"> <li>- 创建管理类追踪器</li> <li>- 示例2：查询追踪器列表</li> <li>- 示例3：查询管理类事件</li> </ul> </li> </ul>
2019-09-03	<p>第十七次正式发布。</p> <p>本次变更说明如下：</p> <p>新增 查询事件列表(v2.0) 章节。</p>
2019-07-24	<p>第十六次正式发布。</p> <p>本次变更说明如下：</p> <p>修改 获取账号ID和项目ID 章节。</p>



发布日期	修订记录
2019-01-30	第十五次正式发布。 本次变更说明如下： 修改 错误码 章节，补充错误码信息。
2018-09-30	第十四次正式发布。 本次变更说明如下： 新增 API版本号管理 章节，支持对API版本号进行查询。
2018-01-30	第十二次正式发布。 本次变更说明如下： <ul style="list-style-type: none"> <li>“关键事件通知”触发条件新增“完整”场景，在创建追踪器、修改追踪器和查询追踪器章节增加相关字段。</li> <li>错误码新增cts.0042~cts.0045和cts.0048。</li> </ul>
2017-11-25	第十一次正式发布。 本次变更说明如下： <ul style="list-style-type: none"> <li>在创建追踪器、修改追踪器章节新增字段，支持事件文件完整性校验功能。</li> <li>在创建追踪器、修改追踪器章节新增字段支持关键事件通知的“用户登录”功能。</li> </ul>
2017-10-30	第十次正式发布。 本次变更说明如下： 在创建追踪器、修改追踪器章节新增字段支持通过LTS对管理类事件进行检索。
2017-09-30	第九次正式发布。 本次变更说明如下： <ul style="list-style-type: none"> <li>增加关键事件通知功能接口描述信息。</li> <li>创建追踪器、修改追踪器和查询追踪器增加是否支持创建桶以及KMS的字段描述。</li> <li>错误码中增加obs错误信息。</li> </ul>
2017-08-30	第八次正式发布。 本次变更说明如下： 在查询事件列表(v1.0)中，新增“request_id”、“location_info”、“endpoint”、“resource_url”字段说明。
2017-05-26	第七次正式发布。 本次变更说明如下： 查询追踪器章节，优化关于接口响应的描述。

发布日期	修订记录
2017-04-28	第六次正式发布。 本次变更说明如下： 事件管理，增加查询事件列表v2.0说明。
2017-02-27	第五次正式发布。 本次变更说明如下： <ul style="list-style-type: none"> <li>• 查询追踪器章节，修改“200”返回值描述。</li> <li>• 在“查询事件列表”接口中，增加“user”字段的说明。</li> </ul>
2017-02-08	第四次正式发布。 本次变更说明如下： <ul style="list-style-type: none"> <li>• 修改追踪器章节，增加“400”返回值。</li> <li>• 查询追踪器章节，增加“400”返回值。</li> <li>• 删除追踪器章节，增加“400”返回值。</li> <li>• 在错误码中，修改cts.0005和cts.0007错误码说明。</li> </ul>
2017-02-03	第三次正式发布。 本次变更说明如下： <ul style="list-style-type: none"> <li>• 创建追踪器章节，增加“400”返回值。</li> <li>• 修改追踪器章节，增加“bucket_name”命名规则，URI的参数说明中增加关于“tracker_name”的说明。</li> <li>• 查询追踪器和删除追踪器章节中，URI的参数说明中增加关于“tracker_name”的说明。删除请求参数中关于“tracker name”的说明。</li> <li>• 错误码中，修改cts.0023错误码说明。</li> <li>• 在“查询事件列表”接口中，修改“trace_id”、“response”、“code”、“message”、“from”、“to”字段说明。增加400和404返回值说明。</li> </ul>
2017-01-20	第二次正式发布。 本次变更说明如下： 查询事件的接口新增servicetype字段。
2016-12-30	第一次正式发布。