Cloud Trace Service

API Reference

Issue	01
Date	2024-05-22

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Before You Start

1.1 Overview

Cloud Trace Service (CTS) is a log audit service designed to strengthen cloud security. It allows you to collect, store, and query resource operation records. You can use these records to perform security analysis, track resource changes, audit compliance, and locate faults.

You can use APIs introduced in this document to perform operations on CTS, such as creating and deleting a tracker. Before calling an API, ensure that you are familiar with related concepts and functions of CTS.

1.2 API Calling

CTS supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see **Calling APIs**.

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of CTS, see **Regions and Endpoints**.

1.4 Constraints

- A maximum of 100 data trackers and 1 management tracker can be created in an account. The quotas cannot be modified.
- For more constraints, see API description.

1.5 Concepts

- Account

  An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.

- User

  An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).

  The account name, username, and password will be required for API authentication.

- Region

  Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

  For details, see **Region and AZ**.

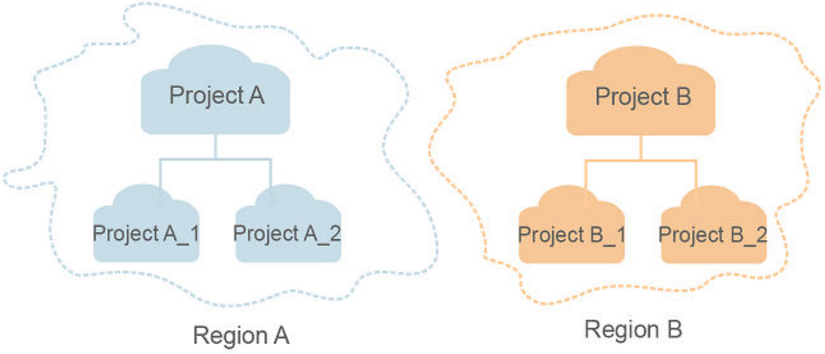
- AZ

  An AZ comprises one or multiple physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

- Project

  A project corresponds to a region. Projects group and isolate resources (including compute, storage, and network resources) across physical regions. Users can be granted permissions in a default project to access all resources in the region associated with the project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



- Enterprise project

  Enterprise projects group and manage resources across regions. Resources in enterprise projects are logically isolated from each other. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.

For details about enterprise projects and how to obtain enterprise project IDs, see **Enterprise Management User Guide**.

1.6 API Versions

It is recommended that you use the V3 APIs, which are more powerful and easy to use. The V1 APIs will be brought offline soon.

2 API Overview

With the extension APIs provided by CTS, you can use all CTS functions, such as querying API versions and the trace list, or creating a tracker.

Table 2-1 lists CTS APIs.

Table 2-1 CTS APIs

Subtype	Description
API versions	API for querying versions of all or specified CTS APIs
Trackers	APIs for creating, modifying, querying, and deleting a tracker
Traces	API for querying traces recorded in the last seven days
Key event notifications	APIs for creating, modifying, querying, and deleting a key event notification

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for **obtaining a user token** as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

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

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

- **URI-scheme**:

  Protocol used to transmit requests. All APIs use **HTTPS**.

- **Endpoint**:

  Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**.

- **resource-path**:

  Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.

- **query-string**:

  Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "**Parameter name=Parameter value**". For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

NOTE

To simplify the URI display, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET**: requests a server to return specified resources.
- **PUT**: requests a server to update specified resources.
- **POST**: requests a server to add resources or perform special operations.
- **DELETE**: requests a server to delete specified resources, for example, objects.
- **HEAD**: same as GET except that the server must return only the response header.
- **PATCH**: requests a server to update a part of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to obtain a user token, the request method is **POST**. The request is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Table 3-1 lists the common request header fields.

Table 3-1 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of *Hostname*:*Port number*. If the port number is not specified, the default port is used. The default port number for **https** is **443**.	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443

Parameter	Description	Mandatory	Example Value
Content-Type	Specifies the type (or format) of the message body. The default value **application/json** is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495
X-Project-Id	Specifies the project ID. Obtain it by referring to **Obtaining an Account ID and Project ID**.	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies a user token. It is a response to the API for **obtaining a user token**. This API is the only one that does not require authentication. After the request is processed, the value of **X-Subject-Token** in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQc-Co...ggg1BBIINPXsidG9rZ

NOTE

In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign a request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

For details, see "AK/SK-based Authentication" in **Authentication**.

The API used to **obtain a user token** does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

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

Request Body (Optional)

This part is optional. A request body transfer information other than the request header and is often sent in a structured format (for example, JSON or XML) defined by the **Content-Type** header field.

A request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to **obtain a user token**, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace ***username***, ***domainname***, ******** (login password), and ***xxxxxxxxxxxxxxxxx*** (project name) with the actual values. Obtain a project name from **Regions and Endpoints**.

NOTE

The **scope** parameter specifies where a token takes effect. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see **Obtaining a User Token**.

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

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

If all data required for the API request is available, you can send the request to call an API through **curl**, **Postman**, or coding. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.2 Authentication

You can use either of the following authentication methods when calling APIs:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. AK/SK-based authentication is recommended because it is more secure than token-based authentication.

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

You can obtain a token by calling the API used for **obtaining a user token**. When you call the API, set **auth.scope** in the request body to **project**.

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

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

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

AK/SK-based Authentication

NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK pair to sign requests based on the signature algorithm or use the signing SDK to sign requests.

NOTICE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

3.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request.

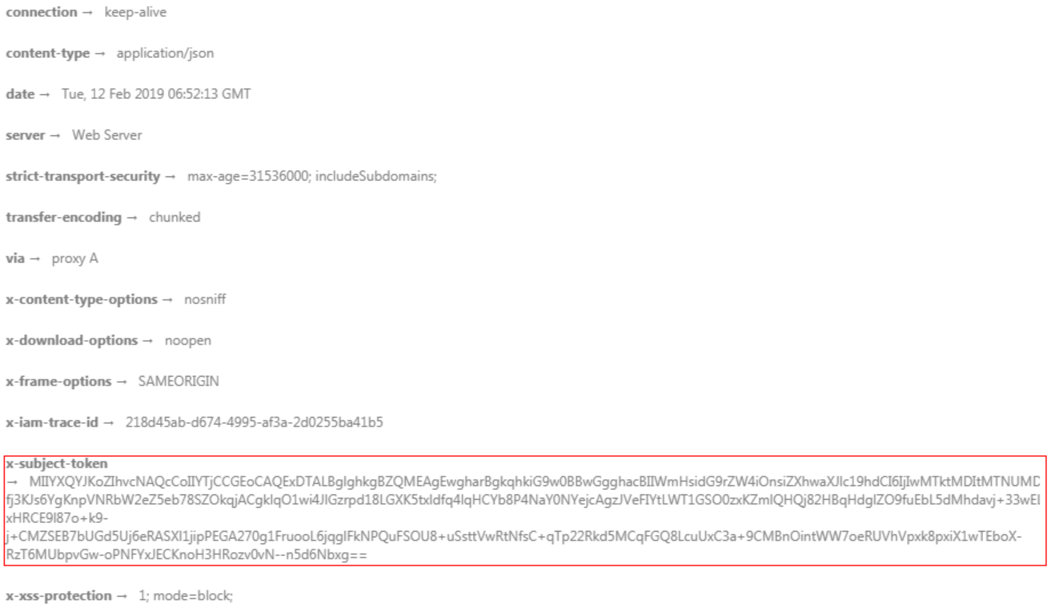
For example, if status code **201** is returned for calling the API used to obtain a user token, the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-type**.

Figure 3-1 shows the response header fields for the API used to obtain a user token. The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-1 Header fields of the response to the request for obtaining a user token



```
connection → keep-alive

content-type → application/json

date → Tue, 12 Feb 2019 06:52:13 GMT

server → Web Server

strict-transport-security → max-age=31536000; includeSubdomains;

transfer-encoding → chunked

via → proxy A

x-content-type-options → nosniff

x-download-options → noopen

x-frame-options → SAMEORIGIN

x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5

x-subject-token
→ MIIYXQYJKoZIhvcNAQcCoIIYTjCCGEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6IjIwMTktMDItMTNUMD
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JIGzrpd18LGXK5txldfq4lqHCYb8P4NaY0NYejcAgzJVeFIYtLWT1GSO0zxKZmIQHQj82HBqHdgIZO9fuEbL5dMhdavj+33wEl
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jqglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==

x-xss-protection → 1; mode=block;
```

Response Body (Optional)

The body of a response is often returned in structured format as specified in the **Content-type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to obtain a user token.

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "az-01",
......
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

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

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 Getting Started

This section describes how to create a tracker by calling APIs.

NOTE

The token obtained from IAM is valid for only 24 hours. If you want to use the same token for authentication, you can cache it to avoid frequent calling of the IAM API.

Involved APIs

For token-based authentication, you must obtain a token and add **X-Auth-Token** to the request header when calling an API.

- IAM API used to obtain a token
- CTS API used to create a tracker

Procedure

1. Obtain a token by referring to **Authentication**.
2. Send **POST /v1.0/ {project_id} /tracker**.
3. Add **Content-Type** and **X-Auth-Token** to the request header.
4. Set parameters in the request body as follows:

```
POST /v1.0/{project_id}/tracker
{
 "bucket_name": "obs-f1da",  //Name of an Object Storage Service (OBS) bucket to which traces will be transferred. This parameter is mandatory and its value is a string.
  "is_support_trace_files_encryption": true,  //Whether trace files will be encrypted during transfer. This parameter is optional and its value is of boolean type.
 "kms_id": "13a4207c-7abe-4b68-8510-16b84c3b5504",  //ID of the key for trace file encryption. When is_support_trace_files_encryption is set to true, this parameter is mandatory.
 "is_obs_created": true,  //Whether a new OBS bucket will be created. This parameter is optional and its value is of boolean type.
 "file_prefix_name": "yO8Q",   //File name prefix to mark trace files that need to be stored in OBS. This parameter is optional and its value is a string.
  "log_file_validate": {
 "is_support_validate": true  //Whether trace files will be verified. This parameter is optional and its value is of boolean type.
 }
}
```

If the request is successful, information about the created tracker is returned.

```
{
 "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",  //Tracker name
"status": "enabled"  //Tracker status
}
```

If the request fails, an error code and error message are returned. For details, see **Error Codes**.

5 Examples

5.1 Example 1: Creating a Management Tracker

Introduction

This section describes how to create a management tracker for an OBS bucket which is used for log transfer.

Other Involved APIs

Listing Buckets: The name of the OBS bucket to which traces will be transferred is required for creating a management tracker. Call this API to list all the OBS buckets you have created to obtain the name.

Creating a Management Tracker

- URI:

  POST /v3/*{project_id}*/tracker

- Example request:

  POST https://{cts_endpoint}/v3/{project_id}/tracker

  Obtain the value of *{cts_endpoint}* from **Regions and Endpoints**.

  Body:

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

- Example response:

```
{
  "id" : "2e6fa9b8-8c6e-456d-b5d3-77be97xxxx",
  "create_time" : 1587958482923,
  "domain_id" : "aexxxxxxxx4d4fb4bexxxxxxx791fbf",
  "obs_info" : {
    "bucket_name" : " test_tracker ",
    "file_prefix_name" : "yO8Q"
  },
  "project_id" : "bb1xxxxxxxxe4f498cbxxxxxxxx35634",
  "tracker_name" : "system",
  "tracker_type" : "system",
  "status" : "enabled"
}
```

5.2 Example 2: Querying a Tracker List

Introduction

This section describes how to query all trackers of an account.

Other Involved APIs

None

Querying a Tracker List

- URI:

  GET /v3/*{project_id}*/trackers

- Example request:

  GET https://{cts_endpoint}/v3/{project_id}/trackers

  Obtain the value of *{cts_endpoint}* from **Regions and Endpoints**.

- Example response:

```
{
    "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
            },
            "create_time": 1589276171198,
            "project_id": "24edf66e79d04187acb99a463e610764",
            "data_bucket": {
                "data_event": [
                    "READ",
                    "WRITE"
                ],
                "search_enabled": false,
                "data_bucket_name": "cstest0423"
            },
            "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 Example 3: Querying Management Traces

Introduction

This section describes how to query all management traces of an account in the last hour.

Other Involved APIs

None

Querying Management Traces

- URI:

  GET /v3/*{project_id}*/traces

- Example request:

  GET https://{cts_endpoint}/v3/{project_id} /traces?trace_type=system

  Obtain the value of *{cts_endpoint}* from **Regions and Endpoints**.

- Example response:

```
{
  "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 V3 APIs (Recommended)

6.1 Key Event Notification Management

6.1.1 Creating a Key Event Notification

Function

SMS, email, or HTTP/HTTPS notifications can be sent through pre-configured SMN topics to subscribers when key operations occur. This helps you detect high-risk operations promptly. Notifications can also be used as triggers for specific operations or to connect to your own audit system.

API Calling

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/notifications

Table 6-1 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Request Parameters

Table 6-2 Request body parameters

Parameter	Mandatory	Type	Description
notification_name	Yes	String	Notification name.
operation_type	Yes	String	Operation type. Possible options include **complete** and **customized**. If you choose **complete**, you do not need to specify **operations** and **notify_user_list**, and notifications will be sent when any supported operations occur on any of the connected cloud services. If you choose **customized**, notifications will be sent when operations defined in **operations** occur. Values: - **complete** - **customized**
operations	No	Array of **Operations** objects	Operation list.
notify_user_list	No	Array of **NotificationUsers** objects	List of users whose operations will trigger notifications. Currently, up to 50 users in 10 user groups can be configured.
topic_id	No	String	Topic URN or function URN. To obtain the **topic_urn**, call the SMN API for querying topics. Example URN: **urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2** To obtain the function URN, call the FunctionGraph API for querying functions. Example URN: **urn:fss:xxxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test**
filter	No	**Filter** object	Advanced filter of key event notifications.

Parameter	Mandatory	Type	Description
agency_name	No	String	Name of a cloud service agency. If this parameter is set to **cts_admin_trust**, a cloud service agency named **cts_admin_trust** is automatically created during key event notification creation. Value: ● **cts_admin_trust**

Table 6-3 Operations

Parameter	Mandatory	Type	Description
service_type	Yes	String	Cloud service. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its acronym.
resource_type	Yes	String	Resource type.
trace_names	Yes	Array of strings	Trace name.

Table 6-4 NotificationUsers

Parameter	Mandatory	Type	Description
user_group	Yes	String	IAM user group.
user_list	Yes	Array of strings	IAM user.

Table 6-5 Filter

Parameter	Mandatory	Type	Description
condition	Yes	String	Relationship between conditions. ● AND (default value) indicates that a rule takes effect after all filtering criteria are met. ● OR indicates that a rule takes effect when one of the filtering criteria is met. Values: ● **AND (default value)** ● **OR**
is_support_filter	Yes	Boolean	Whether to enable the advanced filter.
rule	Yes	Array of strings	Advanced filter criteria rule. Example: **key != value**. The format is *field rule value*. - Field options: **api_version**, **code**, **trace_rating**, **trace_type**, **resource_id**, or **resource_name**. - Rule: **!=** or =. - Value: **api_version**: ^(a-zA-Z0-9_-.){1,64}$; **code**: 1 to 256 characters; **trace_rating**: **normal**, **warning**, or **incident**; **trace_type**: **ConsoleAction**, **ApiCall**, or **SystemAction**; **resource_id**: 1 to 350 characters; **resource_name**: 1 to 256 characters.

Response Parameters

Status code: 201

Table 6-6 Response body parameters

Parameter	Type	Description
notification_name	String	Notification name.

Parameter	Type	Description
operation_type	String	Operation type. Possible options include **complete** and **customized**. - complete: Notifications will be sent through SMN for all operations recorded by CTS. - customized: Notifications will be sent through SMN for specified operations performed on specified cloud services. Values: - **customized** - **complete**
operations	Array of **Operations** objects	Operation list.
notify_user_list	Array of **NotificationUsers** objects	List of users whose operations will trigger notifications. Currently, up to 50 users in 10 user groups can be configured.
status	String	Notification status. - **disabled**: The key event notification is disabled. - **enabled**: The key event notification is enabled. Values: - **enabled** - **disabled**
topic_id	String	Unique resource ID of an SMN topic. You can obtain the ID by querying the topic list.
notification_id	String	Unique notification ID.
notification_type	String	Notification type. - **smn**: Simple Message Notification (SMN). - **fun**: FunctionGraph. Values: - **smn** - **fun**
project_id	String	Project ID.
create_time	Long	Time when a notification rule was created.
filter	**Filter** object	Advanced filter of key event notifications.

Parameter	Type	Description
agency_name	String	Name of a cloud service agency. If this parameter is set to **cts_admin_trust**, a cloud service agency named **cts_admin_trust** is automatically created during key event notification creation. Value: ● **cts_admin_trust**

Table 6-7 Operations

Parameter	Type	Description
service_type	String	Cloud service. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its acronym.
resource_type	String	Resource type.
trace_names	Array of strings	Trace name.

Table 6-8 NotificationUsers

Parameter	Type	Description
user_group	String	IAM user group.
user_list	Array of strings	IAM user.

Table 6-9 Filter

Parameter	Type	Description
condition	String	Relationship between conditions. ● AND (default value) indicates that a rule takes effect after all filtering criteria are met. ● OR indicates that a rule takes effect when one of the filtering criteria is met. Values: ● **AND (default value)** ● **OR**
is_support_filter	Boolean	Whether to enable the advanced filter.
rule	Array of strings	Advanced filter criteria rule. Example: **key != value**. The format is *field rule value*. - Field options: **api_version**, **code**, **trace_rating**, **trace_type**, **resource_id**, or **resource_name**. - Rule: **!=** or **=**. - Value: **api_version**: ^(a-zA-Z0-9_-.){1,64}$; **code**: 1 to 256 characters; **trace_rating**: **normal**, **warning**, or **incident**; **trace_type**: **ConsoleAction**, **ApiCall**, or **SystemAction**; **resource_id**: 1 to 350 characters; **resource_name**: 1 to 256 characters.

Status code: 400

Table 6-10 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-11 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-12 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-13 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-14 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-15 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

- Creating a complete key event notification

```
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"
}
```

- Creating a custom key event notification

```
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"
}
```

Example Response

Status code: 201

The creation is successful.

```
{
  "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"
}
```

Example SDK Code

The example SDK code is as follows.

Java

- Creating a complete key event notification

```
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.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());
        }
    }
}
```

- Creating a custom key event notification

```
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.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

- Creating a complete key event notification

```
# coding: utf-8

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

- Creating a custom key event notification

```
# coding: utf-8

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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("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,
            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

- Creating a complete key event notification

```
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"
    request.Body = &model.CreateNotificationRequestBody{
        Filter: filterbody,
        TopicId: &topicIdCreateNotificationRequestBody,
        OperationType: model.GetCreateNotificationRequestBodyOperationTypeEnum().COMPLETE,
        NotificationName: "test",
    }
    response, err := client.CreateNotification(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Creating a custom key event notification

```
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"
    request.Body = &model.CreateNotificationRequestBody{
        Filter: filterbody,
        TopicId: &topicIdCreateNotificationRequestBody,
        NotifyUserList: &listNotifyUserListbody,
        Operations: &listOperationsbody,
        OperationType: model.GetCreateNotificationRequestBodyOperationTypeEnum().CUSTOMIZED,
        NotificationName: "test",
    }
    response, err := client.CreateNotification(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code in other programming languages, see the **Sample Code** tab in the **API Explorer**.

Status Codes

Status Code	Description
201	The creation is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The server failed to find the requested resource or some key event notifications failed to be deleted.
500	The request failed to be executed or some trackers failed to be deleted.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.1.2 Modifying a Key Event Notification

Function

This API is used to modify a key event notification. The notification ID carried in the request must be valid.

API Calling

For details, see **Calling APIs**.

URI

PUT /v3/{project_id}/notifications

Table 6-16 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Request Parameters

Table 6-17 Request body parameters

Parameter	Mandatory	Type	Description
notification_name	Yes	String	Notification name.
operation_type	Yes	String	Operation type. Possible options include **complete** and **customized**. If you choose **complete**, notifications will be sent when any supported operation occurs on any of the connected cloud services. If you choose **customized**, notifications will be sent when operations defined in **operations** occur. Values: - **customized** - **complete**
operations	No	Array of **Operations** objects	Operation list.
notify_user_list	No	Array of **NotificationUsers** objects	List of users whose operations will trigger notifications. Currently, up to 50 users in 10 user groups can be configured.
status	Yes	String	Notification status. Possible options include **enabled** and **disabled**. Values: - **enabled** - **disabled**

Parameter	Mandatory	Type	Description
topic_id	No	String	Topic URN or function URN. This parameter is mandatory when **status** is set to **enabled**. To obtain the **topic_urn**, call the SMN API for querying topics. Example URN: **urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2** To obtain the function URN, call the FunctionGraph API for querying functions. Example URN: **urn:fss:xxxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test**
notification_id	Yes	String	Notification ID.
filter	No	**Filter** object	Advanced filter of key event notifications.
agency_name	No	String	Name of a cloud service agency. If this parameter is set to **cts_admin_trust**, a cloud service agency named **cts_admin_trust** is automatically created during key event notification creation. Value: ● **cts_admin_trust**

Table 6-18 Operations

Parameter	Mandatory	Type	Description
service_type	Yes	String	Cloud service. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its acronym.

Parameter	Mandatory	Type	Description
resource_type	Yes	String	Resource type.
trace_names	Yes	Array of strings	Trace name.

Table 6-19 NotificationUsers

Parameter	Mandatory	Type	Description
user_group	Yes	String	IAM user group.
user_list	Yes	Array of strings	IAM user.

Table 6-20 Filter

Parameter	Mandatory	Type	Description
condition	Yes	String	Relationship between conditions. ● AND (default value) indicates that a rule takes effect after all filtering criteria are met. ● OR indicates that a rule takes effect when one of the filtering criteria is met. Values: ● **AND (default value)** ● **OR**
is_support_filter	Yes	Boolean	Whether to enable the advanced filter.

Parameter	Mandatory	Type	Description
rule	Yes	Array of strings	Advanced filter criteria rule. Example: **key != value**. The format is *field rule value*. - Field options: **api_version**, **code**, **trace_rating**, **trace_type**, **resource_id**, or **resource_name**. - Rule: **!=** or =. - Value: **api_version**: ^(a-zA-Z0-9_-.){1,64}$; **code**: 1 to 256 characters; **trace_rating**: **normal**, **warning**, or **incident**; **trace_type**: **ConsoleAction**, **ApiCall**, or **SystemAction**; **resource_id**: 1 to 350 characters; **resource_name**: 1 to 256 characters.

Response Parameters

Status code: 200

Table 6-21 Response body parameters

Parameter	Type	Description
notification_name	String	Notification name.
operation_type	String	Operation type. Possible options include **complete** and **customized**. If you choose **complete**, notifications will be sent when any supported operation occurs on any of the connected cloud services. If you choose **customized**, notifications will be sent when operations defined in **operations** occur. Values: ● **customized** ● **complete**
operations	Array of **Operations** objects	Operation list.
notify_user_list	Array of **NotificationUsers** objects	List of users whose operations will trigger notifications. Currently, up to 50 users in 10 user groups can be configured.

Parameter	Type	Description
status	String	Notification status. Possible options include **enabled** and **disabled**. Values: ● **enabled** ● **disabled**
topic_id	String	Topic URN or function URN. To obtain the **topic_urn**, call the SMN API for querying topics. Example URN: **urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2** To obtain the function URN, call the FunctionGraph API for querying functions. Example URN: **urn:fss:xxxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test**
notification_id	String	Unique notification ID.
notification_type	String	Notification type identified based on **topic_id**. Possible options include SMN (**smn**) and FunctionGraph (**fun**). Values: ● **smn** ● **fun**
project_id	String	Project ID.
create_time	Long	Timestamp when a notification was created.
filter	**Filter** object	Advanced filter of key event notifications.
agency_name	String	Name of a cloud service agency. If this parameter is set to **cts_admin_trust**, a cloud service agency named **cts_admin_trust** is automatically created during key event notification modification. Value: ● **cts_admin_trust**

Table 6-22 Operations

Parameter	Type	Description
service_type	String	Cloud service. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its acronym.
resource_type	String	Resource type.
trace_names	Array of strings	Trace name.

Table 6-23 NotificationUsers

Parameter	Type	Description
user_group	String	IAM user group.
user_list	Array of strings	IAM user.

Table 6-24 Filter

Parameter	Type	Description
condition	String	Relationship between conditions. ● AND (default value) indicates that a rule takes effect after all filtering criteria are met. ● OR indicates that a rule takes effect when one of the filtering criteria is met. Values: ● **AND (default value)** ● **OR**
is_support_filter	Boolean	Whether to enable the advanced filter.

Parameter	Type	Description
rule	Array of strings	Advanced filter criteria rule. Example: **key != value**. The format is *field rule value*. - Field options: **api_version**, **code**, **trace_rating**, **trace_type**, **resource_id**, or **resource_name**. - Rule: **!=** or **=**. - Value: **api_version**: ^(a-zA-Z0-9_-.){1,64}$; **code**: 1 to 256 characters; **trace_rating**: **normal**, **warning**, or **incident**; **trace_type**: **ConsoleAction**, **ApiCall**, or **SystemAction**; **resource_id**: 1 to 350 characters; **resource_name**: 1 to 256 characters.

Status code: 400

Table 6-25 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-26 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-27 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-28 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-29 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-30 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

Modify a key event notification.

```
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"
}
```

Example Response

Status code: 200

The notification is modified.

```
{
  "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"
}
```

Status Codes

Status Code	Description
200	The notification is modified.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The server failed to find the requested resource or some key event notifications failed to be deleted.
500	The request failed to be executed or some trackers failed to be deleted.

Status Code	Description
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.1.3 Deleting a Key Event Notification

Function

This API is used to delete a key event notification.

API Calling

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/notifications

Table 6-31 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Table 6-32 Query parameter

Parameter	Mandatory	Type	Description
notification_id	Yes	String	Notification ID. To batch delete notifications, enter multiple notification IDs separated by commas (,), for example, **notification_id="xxx1,ccc2"**.

Request Parameters

None

Response Parameters

Status code: 400

Table 6-33 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-34 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-35 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-36 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-37 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-38 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

None

Example Response

None

Status Codes

Status Code	Description
204	The deletion is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The server failed to find the requested resource or some key event notifications failed to be deleted.
500	The request failed to be executed or some trackers failed to be deleted.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.1.4 Querying a Key Event Notification

Function

This API is used to query a key event notification.

API Calling

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/notifications/{notification_type}

Table 6-39 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.
notification_type	Yes	String	Notification type. - **smn**: Simple Message Notification (SMN). - **fun**: FunctionGraph. Enumerated values: ● **smn** ● **fun**

Table 6-40 Query parameters

Parameter	Mandatory	Type	Description
notification_name	No	String	Notification name. If this parameter is not specified, all key event notifications configured in the current tenant account are returned.

Request Parameters

None

Response Parameters

Status code: 200

Table 6-41 Response body parameters

Parameter	Type	Description
notifications	Array of **NotificationsResponseBody** objects	Notification list.

Table 6-42 NotificationsResponseBody

Parameter	Type	Description
notification_name	String	Notification name.
operation_type	String	Operation type. Possible options include **complete** and **customized**. If you choose **complete**, notifications will be sent when any supported operation occurs on any of the connected cloud services. If you choose **customized**, notifications will be sent when operations defined in **operations** occur. Enumerated values: ● **customized** ● **complete**
operations	Array of **Operations** objects	Operation list.
notify_user_list	Array of **NotificationUsers** objects	List of users whose operations will trigger notifications. Currently, up to 50 users in 10 user groups can be configured.
status	String	Notification status. Possible options include **enabled** and **disabled**. Enumerated values: ● **enabled** ● **disabled**
topic_id	String	Topic URN or function URN. To obtain the **topic_urn**, call the SMN API for querying topics. Example URN: **urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2** To obtain the function URN, call the FunctionGraph API for querying functions. Example URN: **urn:fss:xxxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test**

Parameter	Type	Description
notification_id	String	Unique notification ID.
notification_type	String	Notification type identified based on **topic_id**. Possible options include SMN (**smn**) and FunctionGraph (**fun**). Enumerated values: ● **smn** ● **fun**
project_id	String	Project ID.
create_time	Long	Timestamp when a notification was created.
filter	**Filter** object	Advanced filter of key event notifications.

Table 6-43 Operations

Parameter	Type	Description
service_type	String	Cloud service. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its acronym.
resource_type	String	Resource type.
trace_names	Array of strings	Trace name.

Table 6-44 NotificationUsers

Parameter	Type	Description
user_group	String	IAM user group.
user_list	Array of strings	IAM user.

Table 6-45 Filter

Parameter	Type	Description
condition	String	Relationship between conditions. ● AND (default value) indicates that a rule takes effect after all filtering criteria are met. ● OR indicates that a rule takes effect when one of the filtering criteria is met. Enumerated values: ● **AND (default value)** ● **OR**
is_support_filter	Boolean	Whether to enable the advanced filter.
rule	Array of strings	Advanced filter criteria rule. Example: **key != value**. The format is *field rule value*. - Field options: **api_version**, **code**, **trace_rating**, **trace_type**, **resource_id**, or **resource_name**. - Rule: **!=** or **=**. - Value: **api_version**: ^(a-zA-Z0-9_-.){1,64}$; **code**: 1 to 256 characters; **trace_rating**: **normal**, **warning**, or **incident**; **trace_type**: **ConsoleAction**, **ApiCall**, or **SystemAction**; **resource_id**: 1 to 350 characters; **resource_name**: 1 to 256 characters.

Status code: 400

Table 6-46 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-47 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-48 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-49 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-50 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-51 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

None

Example Response

Status code: 200

The query is successful.

```
{
  "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"
  } ]
}
```

Example SDK Code

The example SDK code is as follows.

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

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

More

For SDK sample code in other programming languages, see the **Sample Code** tab in the **API Explorer**.

Status Codes

Status Code	Description
200	The query is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.

Status Code	Description
403	The server understood the request but refused to authorize it.
404	The server failed to find the requested resource or some key event notifications failed to be deleted.
500	The request failed to be executed or some trackers failed to be deleted.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.2 Trace Management

6.2.1 Querying a Trace List

Function

This API is used to query records of operations on resources in the last seven days.

API Calling

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/traces

Table 6-52 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Table 6-53 Query parameters

Parameter	Mandatory	Type	Description
trace_type	No	String	Trace type. The value can be **system** (management trace) or **data** (data trace) The default value is **system**. Values: ● **system** ● **data**
from	No	Long	UTC millisecond timestamp of the query start time. The value contains 13 digits and the default value is the timestamp of the last hour. Traces generated after the specified timestamp will be queried. The parameters **from** and **to** should be used together.
next	No	String	This parameter is used to query traces generated earlier than its specified value. The value can be that of **marker** in the response. **next** can be used with **from** and **to**. Traces generated in the overlap of the two time ranges specified respectively by **next** and by **from** and **to** will be returned.
to	No	Long	UTC millisecond timestamp of the query end time. The value contains 13 digits and the default value is the timestamp of the current time. Traces generated before the specified timestamp will be queried. The parameters **to** and **from** should be used together.
tracker_name	No	String	When **trace_type** is set to **system**, the default value **system** is used. When **trace_type** is set to **data**, set this parameter to the name of a data tracker to query the traces recorded by this tracker.

Parameter	Mandatory	Type	Description
service_type	No	String	Type of a cloud service whose traces are to be queried. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. This parameter is valid only when **trace_type** is set to **system**. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*.
user	No	String	Name of the user whose traces are to be queried This parameter is valid only when **trace_type** is set to **system**.
resource_name	No	String	Name of a resource whose traces are to be queried. This parameter is valid only when **trace_type** is set to **system**. The value can contain uppercase letters.
resource_type	No	String	Type of a resource whose traces are to be queried. This parameter is valid only when **trace_type** is set to **system**.
resource_id	No	String	ID of a cloud resource whose traces are to be queried. This parameter is valid only when **trace_type** is set to **system**.
limit	No	Integer	Number of traces to query. The default value is **10** and the maximum value is **200**.
trace_id	No	String	Trace ID. If this parameter is specified, other query criteria will not take effect. This parameter is valid only when **trace_type** is set to **system**.
trace_name	No	String	Trace name. This parameter is valid only when **trace_type** is set to **system**. The value can contain uppercase letters.

Parameter	Mandatory	Type	Description
trace_rating	No	String	Trace status. The value can be **normal**, **warning**, or **incident**. This parameter is valid only when **trace_type** is set to **system**. Values: ● **normal** ● **warning** ● **incident**
access_key_id	No	String	ID of the access key used to query traces. It contains temporary access credentials and permanent access keys.
enterprise_project_id	No	String	ID of an enterprise project whose traces are to be queried.

Request Parameters

None

Response Parameters

Status code: 200

Table 6-54 Response body parameters

Parameter	Type	Description
traces	Array of **Traces** objects	List of returned traces.
meta_data	**MetaData** object	Number of returned traces and the marker.

Table 6-55 Traces

Parameter	Type	Description
resource_id	String	ID of a cloud resource on which the recorded operation was performed.

Parameter	Type	Description
trace_name	String	Trace name. The value can contain 1 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). It must start with a letter.
trace_rating	String	Trace status. The value can be **normal**, **warning**, or **incident**. Values: ● **normal** ● **warning** ● **incident**
trace_type	String	Trace source. For management traces, the value can be **ApiCall**, **ConsoleAction**, or **SystemAction**. For data traces, the value can be **ObsSDK** and **ObsAPI**.
request	String	Request of an operation on resources.
response	String	Response to a user request, that is, the returned information for an operation on resources.
code	String	HTTP status code returned by the associated API.
api_version	String	Version of the API.
message	String	Remarks added by other cloud services to a trace.
record_time	Long	Timestamp when an operation was recorded by CTS.
trace_id	String	Trace ID. The value is the UUID generated by the system.
time	Long	Timestamp when a trace was generated.
user	**UserInfo** object	Information of the user who performed the operation that triggered the trace.
service_type	String	Type of a cloud service whose traces are to be queried. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters.
resource_type	String	Type of the resource on which the operation was performed.
source_ip	String	IP address of the tenant who performed the operation that triggered the trace.

Parameter	Type	Description
resource_name	String	Name of a resource on which the recorded operation was performed.
request_id	String	ID of the recorded request.
location_info	String	Additional information required for fault locating after a request error.
endpoint	String	Endpoint in the details page URL of the cloud resource on which the recorded operation was performed.
resource_url	String	Details page URL (excluding the endpoint) of the cloud resource on which the recorded operation was performed.
enterprise_project_id	String	ID of the enterprise project to which the resource belongs.
resource_account_id	String	ID of the account to which the resource belongs. This parameter has a value only when the resource is operated across tenants.

Table 6-56 UserInfo

Parameter	Type	Description
id	String	User ID. For details, see section "Obtaining an Account ID and Project ID" in *Cloud Trace Service API Reference*.
name	String	Username.
user_name	String	Username.
domain	**BaseUser** object	Domain information of the user who performed the operation generating the trace.
account_id	String	Account ID. For details, see section "Obtaining an Account ID and Project ID" in *Cloud Trace Service API Reference*.
access_key_id	String	Access key ID.

Parameter	Type	Description
principal_urn	String	URN of the operator. - For an IAM user, the format is iam::<*account-id*>:user:<*user-name*>. - For an IAM delegated account, the format is sts::sts::<*account-id*>:assumed-agency:<*agency-name*>/<*agency-session-name*>. - For an IAM federated identity, the format is sts::<*account-id*>:external-user:<*idp_id*>/<*user-session-name*>.
principal_id	String	ID of the operator. - For an IAM user, the format is <*user-id*>. - For an IAM delegated account, the format is <*agency-id*>:<*agency-session-name*>. - For an IAM federated identity, the format is <*idp_id*>:<*user-session-name*>.
principal_is_root_user	String	Whether the operator is a root user. - If the value is **true**, the operator is a root user. - If the value is **false**, the operator is an IAM delegated account, federated identity, or non-root user. Values: - **true** - **false**
type	String	Operator identity type.
invoked_by	Array of strings	Name of the service that sends the request. For operations on the console, the value is **["service.console"]**.
session_context	**SessionContext** object	Temporary security credential attribute.

Table 6-57 BaseUser

Parameter	Type	Description
id	String	Account ID. For details, see section "Obtaining an Account ID and Project ID" in *Cloud Trace Service API Reference*.
name	String	Account name.

Table 6-58 SessionContext

Parameter	Type	Description
attributes	**Attributes** object	Temporary security credential attribute.

Table 6-59 Attributes

Parameter	Type	Description
created_at	String	Time when a temporary security credential is issued.
mfa_authenticated	String	Whether MFA identity authentication has been passed.

Table 6-60 MetaData

Parameter	Type	Description
count	Integer	Number of returned traces.
marker	String	ID of the last trace in the returned trace list. The value of this parameter can be used as the **next** value. If the value of **marker** is **null**, all traces have been returned.

Status code: 400

Table 6-61 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-62 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-63 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-64 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-65 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-66 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

- Querying management traces

```
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
```

- Querying data traces

```
GET https://{endpoint}/v3/{project_id}/traces?
limit=11&to=1479095278000&from=1478490478000&trace_type=data
```

Example Response

Status code: 200

The request is successful.

```
{
  "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"
  } ]
}
```

Example SDK Code

The example SDK code is as follows.

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>"));
        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
```

```
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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("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>"
        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
    response, err := client.ListTraces(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code in other programming languages, see the **Sample Code** tab in the **API Explorer**.

Status Codes

Status Codes	Description
200	The request is successful.
400	The request is not completed due to abnormal query parameters.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested trace does not exist.

Status Codes	Description
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.3 Tracker Management

6.3.1 Creating a Tracker

Function

When you enable CTS, a tracker is automatically created to associate with the cloud services you are using and record all operations on the services. A management tracker and multiple data trackers can be created by an account in a region. Traces are retained in the CTS console for seven days. For long-term storage, you can enable Object Storage Service (OBS) and deliver real-time operation records to OBS buckets.

API Calling

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/tracker

Table 6-67 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Request Parameters

Table 6-68 Request body parameters

Parameter	Mandatory	Type	Description
tracker_type	Yes	String	Tracker type. The value can be **system** (management tracker), or **data** (data tracker). Both data and management trackers have the following parameters: **is_lts_enabled**, **obs_info** and **is_support_validate**. Parameters for management trackers: **is_support_trace_files_encryption** and **kms_id**. Parameters for data trackers: **tracker_name** and **data_bucket**. Values: ● **system** ● **data**
tracker_name	Yes	String	Tracker name. When **tracker_type** is set to **system**, the default value **system** is used. When **tracker_type** is set to **data**, you need to set this parameter to a tracker name.
is_lts_enabled	No	Boolean	Whether to enable trace analysis.
is_organization_tracker	No	Boolean	Whether to apply the tracker configuration to the organization. This parameter is valid only for the management tracker. If the value is set to **true**, the audit logs of all members in the ORG organization in the current region will be transferred to the OBS bucket or LTS log stream configured for the management tracker. However, audit logs of other members cannot be viewed on the **Trace List** page.

Parameter	Mandatory	Type	Description
management_event_selector	No	**ManagementEventSelector** object	Management trace selector.
obs_info	No	**TrackerObsInfo** object	Configurations of an OBS bucket to which traces will be transferred.
is_support_trace_files_encryption	No	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter is valid when **tracker_type** is set to **system**. This parameter must be used with **kms_id**.
kms_id	No	String	Key ID used for transferring and encrypting trace files. This key ID is obtained from Key Management Service (KMS). This parameter is valid when **tracker_type** is set to **system**. This parameter is mandatory when **is_support_trace_files_encryption** is set to **true**.
is_support_validate	No	Boolean	Whether trace file verification is enabled for trace transfer.
data_bucket	No	**DataBucket** object	Information of an OBS bucket to be tracked. This parameter is valid when **tracker_type** is set to **data**.
agency_name	No	String	Name of a cloud service agency. If this parameter is set to **cts_admin_trust**, a cloud service agency named **cts_admin_trust** is automatically created during tracker creation. Value: ● **cts_admin_trust**

Table 6-69 ManagementEventSelector

Parameter	Mandatory	Type	Description
exclude_service	No	Array of strings	Cloud services whose traces will not be transferred. Currently, the value can only be set to **KMS**, indicating that the createDatakey traces of KMS will not be transferred.

Table 6-70 TrackerObsInfo

Parameter	Mandatory	Type	Description
bucket_name	No	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
file_prefix_name	No	String	Prefix of trace files that need to be stored in OBS buckets. The value can contain 0 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
is_obs_created	No	Boolean	Whether an OBS bucket is created. If the value is **true**, an OBS bucket will be created to store trace files. If the value is **false**, trace files will be stored in an existing OBS bucket.
bucket_lifecycle	No	Integer	Duration that traces are stored in the OBS bucket. This parameter is valid when **tracker_type** is set to **data**. Values: • **30** • **60** • **90** • **180** • **1,095**

Parameter	Mandatory	Type	Description
compress_type	No	String	Compression type. The value can be JSON (no compression) or GZIP (compression). The default format is GZIP. Values: - **gzip** - **json**
is_sort_by_service	No	Boolean	Whether to sort the path by cloud service. If this option is enabled, the cloud service name is added to the transfer file path. The default value is **true**.

Table 6-71 DataBucket

Parameter	Mandatory	Type	Description
data_bucket_name	No	String	Name of the bucket tracked by a data tracker. - This parameter is mandatory when the data tracker is enabled or disabled. - This parameter is unavailable for a management tracker. - Once a tracker is created, the bucket that it tracks cannot be changed.

Parameter	Mandatory	Type	Description
data_event	No	Array of strings	Type of operations tracked by a data tracker. ● This parameter is mandatory when the data tracker is enabled or disabled. ● This parameter is unavailable for a management tracker. ● **READ**: read operations of an OBS object; **WRITE**: write operations of an OBS object. Values: ● **WRITE** ● **READ**

Response Parameters

Status code: 201

Table 6-72 Response body parameters

Parameter	Type	Description
id	String	Unique tracker ID.
create_time	Long	Timestamp when the tracker was created.
kms_id	String	Key ID used for transferring and encrypting trace files. This key ID is obtained from Key Management Service (KMS). This parameter is mandatory when **tracker_type** is set to **system** and **is_support_trace_files_encryption** is set to **true**.
is_support_validate	Boolean	Whether to enable trace file verification.
is_organization_tracker	Boolean	Whether to apply the tracker configuration to the organization. This parameter is valid only for the management tracker. If the value is set to **true**, the audit logs of all members in the ORG organization in the current region will be transferred to the OBS bucket or LTS log stream configured for the management tracker. However, audit logs of other members cannot be viewed on the **Trace List** page.

Parameter	Type	Description
management_event_selector	**ManagementEventSelector** object	Management trace selector.
lts	**Lts** object	Detail about trace analysis.
tracker_type	String	Tracker type. The value can be **system** (management tracker), or **data** (data tracker). Values: - **system** - **data**
domain_id	String	Account ID. For details, see section "Obtaining an Account ID and Project ID" in *Cloud Trace Service API Reference*.
project_id	String	Project ID.
tracker_name	String	Tracker name. The default value is **system**.
status	String	Tracker status. The value can be **enabled**, **disabled**, or **error**. If the value is set to **error**, the **detail** field is required for specifying the source of the error. Values: - **enabled** - **disabled**
detail	String	This parameter is returned only when the tracker status is **error**. It indicates the cause of the abnormal status, and its value can be **bucketPolicyError**, **noBucket**, or **arrears**.
is_support_trace_files_encryption	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter must be used with **kms_id**. This function is supported only when the value of **tracker_type** is **system**.
obs_info	**ObsInfo** object	Information about the bucket to which traces are transferred.
data_bucket	**DataBucketQuery** object	Information about the bucket tracked by a data tracker. This parameter is valid when **tracker_type** is set to **data**.

Parameter	Type	Description
agency_name	String	Name of a cloud service agency. If this parameter is set to **cts_admin_trust**, a cloud service agency named **cts_admin_trust** is automatically created during tracker creation. Value: ● **cts_admin_trust**

Table 6-73 ManagementEventSelector

Parameter	Type	Description
exclude_service	Array of strings	Cloud services whose traces will not be transferred. Currently, the value can only be set to **KMS**, indicating that the createDatakey traces of KMS will not be transferred.

Table 6-74 Lts

Parameter	Type	Description
is_lts_enabled	Boolean	Specifies whether to enable the LTS search function.
log_group_name	String	Name of the log group that CTS creates in LTS.
log_topic_name	String	Name of the log topic that CTS creates in LTS.

Table 6-75 ObsInfo

Parameter	Type	Description
bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
file_prefix_name	String	Prefix of trace files that need to be stored in OBS buckets. The value can contain 0 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
is_obs_created	Boolean	Whether the OBS bucket is automatically created by the tracker.

Parameter	Type	Description
is_authorized_bucket	Boolean	Whether CTS has been granted permissions to perform operations on the OBS bucket.
bucket_lifecycle	Long	Duration that traces are stored in the OBS bucket. This parameter is valid when **tracker_type** is set to **data**.
compress_type	String	Compression type. The value can be JSON (no compression) or GZIP (compression). The default format is GZIP. Values: ● **gzip** ● **json**
is_sort_by_service	Boolean	Whether to sort the path by cloud service. If this option is enabled, the cloud service name is added to the transfer file path. The default value is **true**.

Table 6-76 DataBucketQuery

Parameter	Type	Description
data_bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
search_enabled	Boolean	Whether the logs of the tracked bucket can be searched.
data_event	Array of strings	Name of the bucket tracked by a data tracker. ● This parameter is mandatory when the data tracker is enabled or disabled. ● This parameter is unavailable for a management tracker. ● Once a tracker is created, the bucket that it tracks cannot be changed. ● **READ OBS**: read operations; **WRITE OBS**: write operations. Values: ● **WRITE** ● **READ**

Status code: 400

Table 6-77 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-78 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-79 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-80 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-81 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-82 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.**_XXX_.
error_msg	String	Error message.

Example Request

- Creating a management tracker

```
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
}
```

- Creating a data tracker

```
{
  "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"
  }
}
```

Example Response

Status code: 201

The request is successful.

```
{
  "id" : "2e6fa9b8-8c6e-456d-b5d3-77be972d220b",
  "create_time" : 1587958482923,
  "domain_id" : "aexxxxxxxx4d4fb4bexxxxxxx791fbf",
  "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"
}
```

Example SDK Code

The example SDK code is as follows.

Java

- Creating a management tracker

```
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.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());
        }
    }
}
```

- Creating a data tracker

```
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.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

- Creating a management tracker

```
# coding: utf-8

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

- Creating a data tracker

```
# coding: utf-8

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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("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="cstest0423",
            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,
            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

- Creating a management tracker

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

- Creating a data tracker

```
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:= "cstest0423"
    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
    request.Body = &model.CreateTrackerRequestBody{
        DataBucket: dataBucketbody,
        ObsInfo: obsInfobody,
        IsLtsEnabled: &isLtsEnabledCreateTrackerRequestBody,
        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)
    }
}
```

More

For SDK sample code in other programming languages, see the **Sample Code** tab in the **API Explorer**.

Status Codes

Status Code	Description
201	The request is successful.
400	The server failed to process the request.

Status Code	Description
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested resource does not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.3.2 Modifying a Tracker

Function

This API is used to modify configurations of a tracker, including trace transfer to OBS buckets, key event notifications, trace file encryption, trace search and analysis using LTS, trace file integrity check, and tracker enablement or disablement. Modifying tracker parameters does not affect the existing operation records. After the modification is complete, the new rules are immediately applied to operation recording.

API Calling

For details, see **Calling APIs**.

URI

PUT /v3/{project_id}/tracker

Table 6-83 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Request Parameters

Table 6-84 Request body parameters

Parameter	Mandatory	Type	Description
tracker_type	Yes	String	Tracker type. The value can be **system** (management tracker), or **data** (data tracker). Both data and management trackers have the following parameters: **is_lts_enabled**, **obs_info** and **is_support_validate**. Parameters for management trackers: **is_support_trace_files_encryption** and **kms_id**. Parameters for data trackers: **tracker_name** and **data_bucket**. Values: - **system** - **data**
tracker_name	Yes	String	Tracker name. When **tracker_type** is set to **system**, the default value **system** is used. When **tracker_type** is set to **data**, you need to set this parameter to a tracker name.
status	No	String	Tracker status. The value can be **enabled** or **disabled**. If you change the value to **disabled**, the tracker stops recording traces. Values: - **enabled** - **disabled**
is_lts_enabled	No	Boolean	Whether to enable trace analysis.

Parameter	Mandatory	Type	Description
is_organization_tracker	No	Boolean	Whether to apply the tracker configuration to the organization. This parameter is valid only for the management tracker. If the value is set to **true**, the audit logs of all members in the ORG organization in the current region will be transferred to the OBS bucket or LTS log stream configured for the management tracker. However, audit logs of other members cannot be viewed on the **Trace List** page.
management_event_selector	No	**ManagementEventSelector** object	Management trace selector.
obs_info	No	**TrackerObsInfo** object	Configurations of an OBS bucket to which traces will be transferred.
is_support_trace_files_encryption	No	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter is valid when **tracker_type** is set to **system**. This parameter must be used with **kms_id**.
kms_id	No	String	Key ID used for transferring and encrypting trace files. This key ID is obtained from Key Management Service (KMS). This parameter is valid when **tracker_type** is set to **system**. This parameter is mandatory when **is_support_trace_files_encryption** is set to **true**.
is_support_validate	No	Boolean	Whether trace file verification is enabled for trace transfer.
data_bucket	No	**DataBucket** object	Information of an OBS bucket to be tracked. This parameter is valid when **tracker_type** is set to **data**.

Parameter	Mandatory	Type	Description
agency_name	No	String	Name of a cloud service agency. If this parameter is set to **cts_admin_trust**, a cloud service agency named **cts_admin_trust** is automatically created during tracker modification. Value: ● **cts_admin_trust**

Table 6-85 ManagementEventSelector

Parameter	Mandatory	Type	Description
exclude_service	No	Array of strings	Cloud services whose traces will not be transferred. Currently, the value can only be set to **KMS**, indicating that the createDatakey traces of KMS will not be transferred.

Table 6-86 TrackerObsInfo

Parameter	Mandatory	Type	Description
bucket_name	No	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
file_prefix_name	No	String	Prefix of trace files that need to be stored in OBS buckets. The value can contain 0 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
is_obs_created	No	Boolean	Whether an OBS bucket is created. If the value is **true**, an OBS bucket will be created to store trace files. If the value is **false**, trace files will be stored in an existing OBS bucket.

Parameter	Mandatory	Type	Description
bucket_lifecycle	No	Integer	Duration that traces are stored in the OBS bucket. This parameter is valid when **tracker_type** is set to **data**. Values: ● **30** ● **60** ● **90** ● **180** ● **1,095**
compress_type	No	String	Compression type. The value can be JSON (no compression) or GZIP (compression). The default format is GZIP. Values: ● **gzip** ● **json**
is_sort_by_service	No	Boolean	Whether to sort the path by cloud service. If this option is enabled, the cloud service name is added to the transfer file path. The default value is **true**.

Table 6-87 DataBucket

Parameter	Mandatory	Type	Description
data_bucket_name	No	String	Name of the bucket tracked by a data tracker. ● This parameter is mandatory when the data tracker is enabled or disabled. ● This parameter is unavailable for a management tracker. ● Once a tracker is created, the bucket that it tracks cannot be changed.

Parameter	Mandatory	Type	Description
data_event	No	Array of strings	Type of operations tracked by a data tracker. ● This parameter is mandatory when the data tracker is enabled or disabled. ● This parameter is unavailable for a management tracker. ● **READ OBS**: read operations; **WRITE OBS**: write operations. Values: ● **WRITE** ● **READ**

Response Parameters

Status code: 400

Table 6-88 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-89 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-90 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-91 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-92 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-93 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

- Modifying a management tracker

```
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"
}
```

- Modifying a data tracker

```
{
  "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" ]
  }
}
```

Example Response

None

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The server failed to find the requested resource.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.3.3 Querying a Tracker

Function

After CTS is enabled, you can view details about the tracker on the **Tracker** page. The details include the name of the tracker, name of the OBS bucket for storing traces, and prefix of the trace files stored in the OBS bucket.

API Calling

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/trackers

Table 6-94 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Table 6-95 Query parameters

Parameter	Mandatory	Type	Description
tracker_name	No	String	Tracker name. If this parameter is not specified, all trackers of a tenant will be queried.
tracker_type	No	String	Tracker type. The value can be **system** (management tracker), or **data** (data tracker). Enumerated values: ● **system** ● **data**

Request Parameters

None

Response Parameters

Status code: 200

Table 6-96 Response body parameters

Parameter	Type	Description
trackers	Array of **TrackerResponseBody** objects	List of tracker information.

Table 6-97 TrackerResponseBody

Parameter	Type	Description
id	String	Unique tracker ID.
create_time	Long	Timestamp when the tracker was created.
kms_id	String	Key ID used for transferring and encrypting trace files. This key ID is obtained from Key Management Service (KMS). This parameter is mandatory when **tracker_type** is set to **system** and **is_support_trace_files_encryption** is set to **true**.
is_support_validate	Boolean	Whether to enable trace file verification.
lts	**Lts** object	Detail about trace analysis.
tracker_type	String	Tracker type. The value can be **system** (management tracker), or **data** (data tracker). Enumerated values: • **system** • **data**
domain_id	String	Account ID. For details, see section "Obtaining an Account ID and Project ID" in *Cloud Trace Service API Reference*.
project_id	String	Project ID.
tracker_name	String	Tracker name. The default value is **system**.
status	String	Tracker status. The value can be **enabled**, **disabled**, or **error**. If the value is set to **error**, the **detail** field is required for specifying the source of the error. Enumerated values: • **enabled** • **disabled**

Parameter	Type	Description
detail	String	This parameter is returned only when the tracker status is **error**. It indicates the cause of the abnormal status, and its value can be **bucketPolicyError**, **noBucket**, or **arrears**.
is_support_trace_files_encryption	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter must be used with **kms_id**. This function is supported only when the value of **tracker_type** is **system**.
obs_info	**ObsInfo** object	Information about the bucket to which traces are transferred.
data_bucket	**DataBucketQuery** object	Information about the bucket tracked by a data tracker. This parameter is valid when **tracker_type** is set to **data**.
group_id	String	LTS log group ID.
stream_id	String	LTS log stream ID.
is_organization_tracker	Boolean	Whether to apply the tracker configuration to the organization. This parameter is valid only for the management tracker. If the value is set to **true**, the audit logs of all members in the ORG organization in the current region will be transferred to the OBS bucket or LTS log stream configured for the management tracker. However, audit logs of other members cannot be viewed on the **Trace List** page.
management_event_selector	**ManagementEventSelector** object	Management trace selector.
agency_name	String	Name of a cloud service agency.

Table 6-98 Lts

Parameter	Type	Description
is_lts_enabled	Boolean	Specifies whether to enable the LTS search function.
log_group_name	String	Name of the log group that CTS creates in LTS.
log_topic_name	String	Name of the log topic that CTS creates in LTS.

Table 6-99 ObsInfo

Parameter	Type	Description
bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
file_prefix_name	String	Prefix of trace files that need to be stored in OBS buckets. The value can contain 0 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
is_obs_created	Boolean	Whether the OBS bucket is automatically created by the tracker.
is_authorized_bucket	Boolean	Whether CTS has been granted permissions to perform operations on the OBS bucket.
bucket_lifecycle	Long	Duration that traces are stored in the OBS bucket. This parameter is valid when **tracker_type** is set to **data**.
compress_type	String	Compression type. The value can be JSON (no compression) or GZIP (compression). The default format is GZIP. Enumerated values: ● **gzip** ● **json**
is_sort_by_service	Boolean	Whether to sort the path by cloud service. If this option is enabled, the cloud service name is added to the transfer file path. The default value is **true**.

Table 6-100 DataBucketQuery

Parameter	Type	Description
data_bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
search_enabled	Boolean	Whether the logs of the tracked bucket can be searched.

Parameter	Type	Description
data_event	Array of strings	Name of the bucket tracked by a data tracker. ● This parameter is mandatory when the data tracker is enabled or disabled. ● This parameter is unavailable for a management tracker. ● Once a tracker is created, the bucket that it tracks cannot be changed. ● **READ OBS**: read operations; **WRITE OBS**: write operations. Enumerated values: ● **WRITE** ● **READ**

Table 6-101 ManagementEventSelector

Parameter	Type	Description
exclude_service	Array of strings	Cloud services whose traces will not be transferred. Currently, the value can only be set to **KMS**, indicating that the createDatakey traces of KMS will not be transferred.

Status code: 400

Table 6-102 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-103 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-104 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-105 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-106 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

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

Example Response

Status code: 200

The request is successful.

```
{
  "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" : "cstest0423"
    },
    "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"
  } ]
}
```

Example SDK Code

The example SDK code is as follows.

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

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

More

For SDK sample code in other programming languages, see the **Sample Code** tab in the **API Explorer**.

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
500	Failed to complete the request because of an internal service error.

Status Code	Description
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.3.4 Deleting a Tracker

Function

This API is used to delete a tracker. Only data trackers can be deleted. Deleting a tracker has no impact on the operation records that have been generated. When you enable CTS again, you can still view those traces.

API Calling

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/trackers

Table 6-107 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Table 6-108 Query parameters

Parameter	Mandatory	Type	Description
tracker_name	No	String	Tracker name. If this parameter is not specified, all data trackers of the current tenant account will be deleted.
tracker_type	No	String	Tracker type. Only data trackers can be deleted. The default value is **data**. Value: - **data**

Request Parameters

None.

Response Parameters

Status code: 400

Table 6-109 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-110 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-111 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-112 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-113 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-114 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

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

Example Response

None.

Example SDK Code

The example SDK code is as follows.

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

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

More

For SDK sample code in other programming languages, see the **Sample Code** tab in the **API Explorer**.

Status Codes

Status Code	Description
204	The deletion is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The server failed to find the requested resource or some trackers failed to be deleted.
500	The request failed to be executed or some trackers failed to be deleted.

Status Code	Description
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.4 Other APIs

6.4.1 Querying the Tracker Quota of a Tenant

Function

This API is used to query the tracker quota of a tenant.

URI

GET /v3/{project_id}/quotas

Table 6-115 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Request Parameters

None

Response Parameters

Status code: 200

Table 6-116 Response body parameter

Parameter	Type	Description
resources	Array of **Quota** objects	List of tracker information.

Table 6-117 Quota

Parameter	Type	Description
type	String	Quota resource type.
used	Long	Number of used resources.
quota	Long	Total number of resources.

Status code: 400

Table 6-118 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-119 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-120 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-121 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.

Parameter	Type	Description
error_msg	String	Error message.

Status code: 500

Table 6-122 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-123 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

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

Example Response

Status code: 200

The request is successful.

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

Status Code

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested resource does not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Code

For details, see **Error Codes**.

6.4.2 Querying All Operations on a Cloud Service

Function

This API is used to list all operations on a cloud service.

URI

GET /v3/{project_id}/operations

Table 6-124 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Table 6-125 Query parameters

Parameter	Mandatory	Type	Description
service_type	No	String	Type of the cloud service on which operations are performed.

Parameter	Mandatory	Type	Description
resource_type	No	String	Type of the resource on which operations are performed. If this parameter is used, **service_type** is mandatory.

Request Parameters

None

Response Parameters

Status code: 200

Table 6-126 Response body parameter

Parameter	Type	Description
operations	Array of **ListOperation** objects	All operations on the cloud service.

Table 6-127 ListOperation

Parameter	Type	Description
service_type	String	Type of the cloud service on which operations are performed.
resource_type	String	Type of the resource on which operations are performed.
operation_list	Array of strings	Array of operation names.

Status code: 400

Table 6-128 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-129 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-130 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-131 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-132 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-133 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

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

Example Response

Status code: 200

The request is successfully sent.

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

Status Codes

Status Code	Description
200	The request is successfully sent.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested resource does not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Codes

For details, see **Error Codes**.

6.4.3 Querying the Audit Log Operator List

Function

This API is used to query the audit log operator list.

URI

GET /v3/{project_id}/user-resources

Table 6-134 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.

Request Parameters

None

Response Parameters

Status code: 200

Table 6-135 Response body parameter

Parameter	Type	Description
users	Array of **UserResource** objects	List of users who have performed operations in the last 30 days.

Table 6-136 UserResource

Parameter	Type	Description
name	String	Username.

Status code: 400

Table 6-137 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-138 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-139 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-140 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-141 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-142 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

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

Example Response

Status code: 200

The request is successfully sent.

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

Status Codes

Status Code	Description
200	The request is successfully sent.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested resource does not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Codes

For details, see **Error Codes**.

6.4.4 Checking Whether Data Can Be Transferred to the Configured OBS Bucket

Function

This API is used to check whether data can be transferred to the OBS bucket.

URI

POST /v3/{domain_id}/checkbucket

Table 6-143 Path parameter

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Account ID. For details, see **Obtaining an Account ID and Project ID**.

Request Parameters

Table 6-144 Request body parameter

Parameter	Mandatory	Type	Description
buckets	No	Array of **CheckBucketRequest** objects	List of OBS buckets to be checked.

Table 6-145 CheckBucketRequest

Parameter	Mandatory	Type	Description
bucket_name	Yes	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
bucket_location	Yes	String	OBS bucket location.
kms_id	No	String	Key ID used for encrypting transferred trace files. This parameter is mandatory when **is_support_trace_files_encryption** is set to **true**.
is_support_trace_files_encryption	No	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter must be used with **kms_id**.

Response Parameters

Status code: 200

Table 6-146 Response body parameter

Parameter	Type	Description
buckets	Array of **Bucket** objects	Response body for checking the OBS bucket status.

Table 6-147 Bucket

Parameter	Type	Description
bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
bucket_location	String	Bucket location.
kms_id	String	Key ID used for transferring and encrypting trace files. This parameter is mandatory when **is_support_trace_files_encryption** is set to **true**.
is_support_trace_files_encryption	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter must be used with **kms_id**.
check_bucket_response	**CheckBucketResponse** object	Check result of the OBS bucket.

Table 6-148 CheckBucketResponse

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.
response_code	Integer	Returned HTTP status code.
success	Boolean	Whether the transfer is successful.

Status code: 400

Table 6-149 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-150 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-151 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-152 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-153 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-154 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

Example request body for checking the OBS bucket status:

```
GET 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"
  } ]
}
```

Example Response

Status code: 200

The request is successfully sent.

```
{
  "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"
  } ]
}
```

Status Codes

Status Code	Description
200	The request is successfully sent.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested resource does not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Codes

For details, see **Error Codes**.

6.4.5 Querying the Resources Involved in the Traces

Function

This API is used to query the resources involved in the traces.

URI

GET /v3/{domain_id}/resources

Table 6-155 Path parameter

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Account ID. For details, see **Obtaining an Account ID and Project ID**.

Request Parameters

None

Response Parameters

Status code: 200

Table 6-156 Response body parameter

Parameter	Type	Description
resources	Array of **TraceResource** objects	Returned resource list.

Table 6-157 TraceResource

Parameter	Type	Description
service_type	String	Cloud service type. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters.
resource	Array of strings	Resources corresponding to the cloud services.

Status code: 400

Table 6-158 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 401

Table 6-159 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-160 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.

Parameter	Type	Description
error_msg	String	Error message.

Status code: 404

Table 6-161 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-162 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-163 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

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

Example Response

Status code: 200

The request is successfully sent.

```
{
  "resources" : [ {
    "service_type" : "CTS",
    "resource" : [ "trace", "notification", "tracker" ]
  }, {
```

```
  "service_type" : "AOM",
  "resource" : [ "xxxx1", "xxxx2" ]
 } ]
}
```

Status Codes

Status Code	Description
200	The request is successfully sent.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested resource does not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Codes

For details, see **Error Codes**.

6.5 Tag Management

6.5.1 Adding CTS Resource Tags in Batches

Function

This API is used to add CTS resource tags in batches.

URI

POST /v3/{project_id}/{resource_type}/{resource_id}/tags/create

Table 6-164 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.
resource_id	Yes	String	Resource ID.

Parameter	Mandatory	Type	Description
resource_type	Yes	String	Resource type of CTS. Currently, only **cts-tracker** is supported. Value: ● **cts-tracker**

Request Parameters

Table 6-165 Request body parameter

Parameter	Mandatory	Type	Description
tags	No	Array of **Tags** objects	List of tags.

Table 6-166 Tags

Parameter	Mandatory	Type	Description
key	No	String	Tag key, which can contain a maximum of 128 Unicode characters. A tag value can contain letters, digits, spaces, and special characters (_.:/=+-@). It cannot start or end with a space, or start with _sys_.
value	No	String	Tag value, which can contain a maximum of 255 Unicode characters. If **value** is specified, tags are deleted by key and value. If **value** is not specified, tags are deleted by key. A tag value can contain letters, digits, spaces, and special characters (_.:/=+-@) but cannot start or end with a space.

Response Parameters

Status code: 401

Table 6-167 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-168 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-169 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-170 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-171 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

Example of creating a tracker tag:

```
POST https://{endpoint}/v3/{project_id}/{resource_type}/{resource_id}/tags/create

{
  "tags" : [ {
    "key" : "111",
    "value" : "33"
  } ]
}
```

Example Response

None

Status Codes

Status Code	Description
200	The creation is successful.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested resource does not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Codes

For details, see **Error Codes**.

6.5.2 Deleting CTS Resource Tags in Batches

Function

This API is used to delete CTS resource tags in batches.

URI

DELETE /v3/{project_id}/{resource_type}/{resource_id}/tags/delete

Table 6-172 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see **Obtaining an Account ID and Project ID**.
resource_id	Yes	String	Resource ID.
resource_type	Yes	String	Resource type of CTS. Currently, only cts-tracker is supported. Value: ● **cts-tracker**

Request Parameters

Table 6-173 Request body parameter

Parameter	Mandatory	Type	Description
tags	No	Array of **Tags** objects	List of tags.

Table 6-174 Tags

Parameter	Mandatory	Type	Description
key	No	String	Tag key, which can contain a maximum of 128 Unicode characters. A tag value can contain letters, digits, spaces, and special characters (_.:/=+-@). It cannot start or end with a space, or start with _sys_.
value	No	String	Tag value, which can contain a maximum of 255 Unicode characters. If **value** is specified, tags are deleted by key and value. If **value** is not specified, tags are deleted by key. A tag value can contain letters, digits, spaces, and special characters (_.:/=+-@) but cannot start or end with a space.

Response Parameters

Status code: 401

Table 6-175 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 403

Table 6-176 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 404

Table 6-177 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 500

Table 6-178 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Status code: 503

Table 6-179 Response body parameters

Parameter	Type	Description
error_code	String	Error code. Format: **CTS.***XXX*.
error_msg	String	Error message.

Example Request

Example of deleting a tracker tag:

```
DELETE https://{endpoint}/v3/{project_id}/{resource_type}/{resource_id}/tags/delete

{
  "tags" : [ {
    "key" : "111",
    "value" : "33"
  } ]
}
```

Example Response

None

Status Codes

Status Code	Description
204	The deletion is successful.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested resource does not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Codes

For details, see **Error Codes**.

7 Out-of-Date APIs

7.1 API Version Management

7.1.1 Querying All API Versions

Function

This API is used to query all API versions supported by CTS.

URI

GET /

Request Parameters

None

Response Parameters

Status code: 200

Table 7-1 Response body parameters

Parameter	Type	Description
versions	Array of **Versions** objects	List of all versions.

Table 7-2 Versions

Parameter	Type	Description
id	String	Specifies the version ID, for example, v1.
links	Array of **LinksBody** objects	Specifies the API URL.
version	String	If microversions are supported, set this parameter to the maximum microversion supported. Otherwise, leave this parameter empty.
status	String	Specifies the version status. **CURRENT**: widely used version. **SUPPORTED**: earlier version that is still supported. **DEPRECATED**: deprecated version which may be deleted later. Value: ● **CURRENT** ● **SUPPORTED** ● **DEPRECATED**
updated	String	The version release time in UTC. For example, the release time of v1 is **2014-06-28T12:20:21Z**.
min_version	String	If microversions are supported, set this parameter to the minimum microversion supported. Otherwise, leave this parameter empty.

Table 7-3 LinksBody

Parameter	Type	Description
href	String	Specifies the reference address of the current API version.
rel	String	Specifies the relationship between the current API version and the referenced address.

Example Request

None

Example Response

Status code: 200

The request is successful.

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

Status Codes

Status Code	Description
200	The request is successful.
404	The server failed to find the requested resource.
500	The server encountered an unexpected condition which prevented it from fulfilling the request.

Error Codes

For details, see **Error Codes**.

7.1.2 Querying a Specific API Version

Function

This API is used to query a specified API version of CTS.

URI

GET /{version}

Table 7-4 Path parameter

Parameter	Mandatory	Type	Description
version	Yes	String	Version number.

Request Parameters

None

Response Parameters

Status code: 200

Table 7-5 Response body parameters

Parameter	Type	Description
version	Object	Information of the version.

Table 7-6 Version

Parameter	Type	Description
id	String	Specifies the version ID, for example, v1.
links	Array of **LinksBody** objects	Specifies the API URL.
version	String	If microversions are supported, set this parameter to the maximum microversion supported. Otherwise, leave this parameter empty.
status	String	Specifies the version status. **CURRENT**: widely used version. **SUPPORTED**: earlier version that is still supported. **DEPRECATED**: deprecated version which may be deleted later. Value: ● **CURRENT** ● **SUPPORTED** ● **DEPRECATED**
updated	String	The version release time in UTC. For example, the release time of v1 is **2014-06-28T12:20:21Z**.
min_version	String	If microversions are supported, set this parameter to the minimum microversion supported. Otherwise, leave this parameter empty.

Table 7-7 LinksBody

Parameter	Type	Description
href	String	Specifies the reference address of the current API version.
rel	String	Specifies the relationship between the current API version and the referenced address.

Example Request

None

Example Response

Status code: 200

The request is successful.

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

Status Codes

Status Code	Description
200	The request is successful.
404	The server failed to find the requested resource.
500	The server encountered an unexpected condition which prevented it from fulfilling the request.

Error Codes

For details, see **Error Codes**.

7.2 Trace Management

7.2.1 Querying Traces (v1.0)

Function

This API is used to query records of operations on resources in the last seven days.

URI

GET /v1.0/{project_id}/{tracker_name}/trace

Table 7-8 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID.
tracker_name	Yes	String	Tracker name. Currently, only one tracker is allowed for each tenant. The tracker name is **system**.

Table 7-9 Query parameters

Parameter	Mandatory	Type	Description
service_type	No	String	Type of a cloud service whose traces are to be queried. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its name abbreviation.
user	No	String	Name of the user whose traces are to be queried.

Parameter	Mandatory	Type	Description
from	No	Long	UTC millisecond timestamp of the query start time. The value contains 13 digits and the default value is the timestamp of the last hour. Traces generated after the specified timestamp will be queried. The parameters **from** and **to** should be used together.
limit	No	Integer	Number of traces to query. The default value is **10** and the maximum value is **200**.
next	No	String	This parameter is used to query traces generated earlier than its specified value. The value can be that of **marker** in the response. **next** can be used with **from** and **to**. Traces generated in the overlap of the two time ranges specified respectively by **next** and by **from** and **to** will be returned.
resource_id	No	String	ID of a cloud resource whose traces are to be queried.
resource_name	No	String	Name of a resource whose traces are to be queried. The value can contain uppercase letters.
resource_type	No	String	Type of a resource whose traces are to be queried.
to	No	Long	UTC millisecond timestamp of the query end time. The value contains 13 digits and the default value is the timestamp of the current time. Traces generated before the specified timestamp will be queried. The parameters **to** and **from** should be used together.
trace_id	No	String	Trace ID. If this parameter is specified, other query criteria will not take effect.
trace_name	No	String	Trace name. The value can contain uppercase letters.

Parameter	Mandatory	Type	Description
trace_rating	No	String	Trace status. The value can be **normal**, **warning**, or **incident**.

Request Parameters

None

Response Parameters

Status code: 200

Table 7-10 Response body parameters

Parameter	Type	Description
traces	Array of **Traces** objects	List of returned traces.
meta_data	**MetaData** object	Number of returned traces and the marker.

Table 7-11 Traces

Parameter	Type	Description
resource_id	String	ID of a cloud resource on which the recorded operation was performed.
trace_name	String	Trace name. The value can contain 1 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). It must start with a letter.
trace_rating	String	Trace status. The value can be **normal**, **warning**, or **incident**. Value: • **normal** • **warning** • **incident**
trace_type	String	Trace source. The value can be **ApiCall**, **ConsoleAction**, or **SystemAction**.
request	String	Request of an operation on resources.
response	String	Response to a user request, that is, the returned information for an operation on resources.

Parameter	Type	Description
code	String	HTTP status code returned by the associated API.
api_version	String	Version of the API called in a trace.
message	String	Remarks added by other cloud services to a trace.
record_time	Long	Timestamp when a trace was recorded by CTS.
trace_id	String	Trace ID. The value is the UUID generated by the system.
time	Long	Timestamp when a trace was generated.
user	**UserInfo** object	Information of the user who performed the operation that triggered the trace.
service_type	String	Type of a cloud service whose traces are to be queried. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its name abbreviation.
resource_type	String	Type of the resource on which the operation was performed.
source_ip	String	IP address of the tenant who performed the operation that triggered the trace.
resource_name	String	Name of a resource on which the recorded operation was performed.
request_id	String	Request ID.
location_info	String	Additional information required for fault locating after a request error.
endpoint	String	Endpoint in the details page URL of the cloud resource on which the recorded operation was performed.
resource_url	String	Details page URL (excluding the endpoint) of the cloud resource on which the recorded operation was performed.

Table 7-12 UserInfo

Parameter	Type	Description
id	String	Account ID. For details about how to obtain an account ID, see Obtaining an Account ID.
name	String	Account name.
domain	**BaseUser** object	Domain information of the user who performed the operation generating the trace.

Table 7-13 BaseUser

Parameter	Type	Description
id	String	Account ID. For details about how to obtain an account ID, see Obtaining an Account ID.
name	String	Account name.

Table 7-14 MetaData

Parameter	Type	Description
count	Integer	Number of returned traces.
marker	String	ID of the last trace in the returned trace list. The value of this parameter can be used as the value of **next**. If the value of **marker** is **null**, all traces have been returned under the specified query criteria.

Example Request

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

Example Response

Status code: 200

The request is successful.

```
{
  "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"
  } ]
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The request is not completed due to abnormal query parameters.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested traces do not exist.
500	Failed to complete the request because of an internal service error.

Status Code	Description
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Codes

For details, see **Error Codes**.

7.2.2 Querying Traces (v2.0)

Function

This API is used to query records of operations on resources in the last seven days.

URI

GET /v2.0/{project_id}/{tracker_name}/trace

Table 7-15 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID.
tracker_name	Yes	String	Tracker name. Currently, only one tracker is allowed for each tenant. The tracker name is **system**.

Table 7-16 Query parameters

Parameter	Mandatory	Type	Description
service_type	No	String	Type of a cloud service whose traces are to be queried. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its name abbreviation.
user	No	String	Name of the user whose traces are to be queried.
from	No	Long	UTC millisecond timestamp of the query start time. The value contains 13 digits and the default value is the timestamp of the last hour. Traces generated after the specified timestamp will be queried. The parameters **from** and **to** should be used together.
limit	No	Integer	Number of traces to query. The default value is **10** and the maximum value is **200**.
next	No	String	This parameter is used to query traces generated earlier than its specified value. The value can be that of **marker** in the response. **next** can be used with **from** and **to**. Traces generated in the overlap of the two time ranges specified respectively by **next** and by **from** and **to** will be returned.
resource_id	No	String	ID of a cloud resource whose traces are to be queried.
resource_name	No	String	Name of a resource whose traces are to be queried. The value can contain uppercase letters.

Parameter	Mandatory	Type	Description
resource_type	No	String	Type of a resource whose traces are to be queried.
to	No	Long	UTC millisecond timestamp of the query end time. The value contains 13 digits and the default value is the timestamp of the current time. Traces generated before the specified timestamp will be queried. The parameters **to** and **from** should be used together.
trace_id	No	String	Trace ID. If this parameter is specified, other query criteria will not take effect.
trace_name	No	String	Trace name. The value can contain uppercase letters.
trace_rating	No	String	Trace status. The value can be **normal**, **warning**, or **incident**.

Request Parameters

None

Response Parameters

Status code: 200

Table 7-17 Response body parameters

Parameter	Type	Description
traces	Array of **Traces** objects	List of returned traces.
meta_data	**MetaData** object	Number of returned traces and the marker.

Table 7-18 Traces

Parameter	Type	Description
resource_id	String	ID of a cloud resource on which the recorded operation was performed.

Parameter	Type	Description
trace_name	String	Trace name. The value can contain 1 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). It must start with a letter.
trace_rating	String	Trace status. The value can be **normal**, **warning**, or **incident**. Value: ● **normal** ● **warning** ● **incident**
trace_type	String	Trace source. The value can be **ApiCall**, **ConsoleAction**, or **SystemAction**.
request	String	Request of an operation on resources.
response	String	Response to a user request, that is, the returned information for an operation on resources.
code	String	HTTP status code returned by the associated API.
api_version	String	Version of the API called in a trace.
message	String	Remarks added by other cloud services to a trace.
record_time	Long	Timestamp when a trace was recorded by CTS.
trace_id	String	Trace ID. The value is the UUID generated by the system.
time	Long	Timestamp when a trace was generated.
user	**UserInfo** object	Information of the user who performed the operation that triggered the trace.
service_type	String	Type of a cloud service whose traces are to be queried. The value must be the acronym of a cloud service that has been connected with CTS. It is a word composed of uppercase letters. For cloud services that can be connected with CTS, see section "Supported Services and Operations" in *Cloud Trace Service User Guide*. You can click the document link of each cloud service to view its name abbreviation.
resource_type	String	Type of the resource on which the operation was performed.

Parameter	Type	Description
source_ip	String	IP address of the tenant who performed the operation that triggered the trace.
resource_name	String	Name of a resource on which the recorded operation was performed.
request_id	String	Request ID.
location_info	String	Additional information required for fault locating after a request error.
endpoint	String	Endpoint in the details page URL of the cloud resource on which the recorded operation was performed.
resource_url	String	Details page URL (excluding the endpoint) of the cloud resource on which the recorded operation was performed.

Table 7-19 UserInfo

Parameter	Type	Description
id	String	Account ID. For details about how to obtain an account ID, see Obtaining an Account ID.
name	String	Account name.
domain	**BaseUser** object	Domain information of the user who performed the operation generating the trace.

Table 7-20 BaseUser

Parameter	Type	Description
id	String	Account ID. For details about how to obtain an account ID, see Obtaining an Account ID.
name	String	Account name.

Table 7-21 MetaData

Parameter	Type	Description
count	Integer	Number of returned traces.

Parameter	Type	Description
marker	String	ID of the last trace in the returned trace list. The value of this parameter can be used as the value of **next**. If the value of **marker** is **null**, all traces have been returned under the specified query criteria.

Example Request

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

Example Response

Status code: 200

The request is successful.

```
{
  "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"
  } ]
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The request is not completed due to abnormal query parameters.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested traces do not exist.
500	Failed to complete the request because of an internal service error.
503	The requested service is invalid. The client should not repeat the request without modifications.

Error Codes

For details, see **Error Codes**.

7.3 Tracker Management

7.3.1 Creating a Tracker

Function

When you enable CTS, a tracker is automatically created to associate with the cloud services you are using and record all operations on the services. Currently, only one management tracker can be created for a cloud account in a region. Traces are retained in the CTS console for seven days. For long-term storage, you can enable Object Storage Service (OBS) and deliver real-time operation records to OBS buckets.

URI

POST /v1.0/{project_id}/tracker

Table 7-22 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID.

Request Parameters

Table 7-23 Request body parameters

Parameter	Mandatory	Type	Description
bucket_name	Yes	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
file_prefix_name	No	String	Prefix of trace files that need to be stored in OBS buckets. The value can contain 0 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
is_obs_created	No	Boolean	Whether an OBS bucket is created. The default value is **false**. If the value is **true**, an OBS bucket will be created to store trace files. If the value is **false**, trace files will be stored in an existing OBS bucket. A bucket name contains 3 to 64 characters, including digits, letters, and a hyphen (-) or a period (.).
is_support_trace_files_encryption	No	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter must be used with **kms_id**.

Parameter	Mandatory	Type	Description
kms_id	No	String	Key ID used for transferring and encrypting trace files. This key ID is obtained from Key Management Service (KMS). This parameter is mandatory when **is_support_trace_files_encryption** is set to **true**.
lts	No	**Lts** object	Trace analysis.
log_file_validate	No	**LogFileValidate** object	File verification.

Table 7-24 Lts

Parameter	Mandatory	Type	Description
is_lts_enabled	Yes	Boolean	Whether the LTS search function is enabled.
log_group_name	Yes	String	Name of the log group that CTS creates in LTS.
log_topic_name	Yes	String	Name of the log topic that CTS creates in LTS.

Table 7-25 LogFileValidate

Parameter	Mandatory	Type	Description
is_support_validate	Yes	Boolean	Whether to enable trace file verification.

Response Parameters

Status code: 201

Table 7-26 Response body parameters

Parameter	Type	Description
id	String	Unique tracker ID.
create_time	Long	Timestamp when the tracker was created.

Parameter	Type	Description
domain_id	String	Account ID. For details about how to obtain an account ID, see Obtaining an Account ID.
project_id	String	Project ID.
is_support_trace_files_encryption	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter must be used with **kms_id**.
kms_id	String	Key ID used for transferring and encrypting trace files. This key ID is obtained from Key Management Service (KMS). This parameter is mandatory when **is_support_trace_files_encryption** is set to **true**.
obs_info	**ObsInfo** object	Information about the bucket to which traces are transferred.
status	String	Status of a tracker. The value **enabled** indicates normal status.
tracker_name	String	Tracker name.
tracker_type	String	Tracker type.
group_id	String	Unique ID of a log group.
stream_id	String	Unique ID of a log stream.
lts	**Lts** object	Trace analysis.
is_support_validate	Boolean	Whether to enable trace file verification. This function is supported only when the value of **tracker_type** is **system**.

Table 7-27 ObsInfo

Parameter	Type	Description
bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
file_prefix_name	String	Prefix of trace files that need to be stored in OBS buckets. The value can contain 0 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
is_obs_created	Boolean	Whether the OBS bucket is automatically created by the tracker.

Parameter	Type	Description
is_authorized_bucket	Boolean	Whether CTS has been granted permissions to perform operations on the OBS bucket.
bucket_lifecycle	String	Duration that traces are stored in the OBS bucket. This parameter is valid when **tracker_type** is set to **data**.

Table 7-28 Lts

Parameter	Type	Description
is_lts_enabled	Boolean	Whether the LTS search function is enabled.
log_group_name	String	Name of the log group that CTS creates in LTS.
log_topic_name	String	Name of the log topic that CTS creates in LTS.

Example Request

```
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
 }
}
```

Example Response

Status code: 201

The request is successfully sent.

```
{
  "id" : "2e6fa9b8-8c6e-456d-b5d3-77be972d220b",
  "create_time" : 1587958482923,
  "domain_id" : "aexxxxxxxxx4d4fb4bexxxxxxx791fbf",
  "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" : "bb1xxxxxxxxe4f498cbxxxxxxxx35634",
  "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"
}
```

Status Codes

Status Code	Description
201	The request is successfully sent.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The requested OBS bucket does not exist.
500	Failed to complete the request because of an internal service error.

Error Codes

For details, see **Error Codes**.

7.3.2 Querying a Tracker

Function

This API is used to query the details of a tracker. The details include the name of the tracker, name of the OBS bucket for storing traces, and prefix of the trace files stored in the OBS bucket.

URI

GET /v1.0/{project_id}/tracker

Table 7-29 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID.

Table 7-30 Query parameter

Parameter	Mandatory	Type	Description
tracker_name	Yes	String	Tracker name. If this parameter is not specified, all trackers of a tenant will be queried. Currently, only one tracker is allowed for each tenant. The tracker name is **system**.

Request Parameters

None

Response Parameters

Status code: 200

Table 7-31 Response body parameters

Parameter	Type	Description
id	String	Unique tracker ID.
create_time	Long	Timestamp when the tracker was created.
domain_id	String	Account ID. For details about how to obtain an account ID, see Obtaining an Account ID.
project_id	String	Project ID.
tracker_name	String	Tracker name. The default value is **system**.
tracker_type	String	Tracker type.
bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
file_prefix_name	String	Prefix of trace files that need to be stored in OBS buckets.
status	String	Tracker status. The value can be **enabled**, **disabled**, or **error**. If the value is set to **error**, the **detail** field is required for specifying the source of the error. Value: ● **enabled** ● **disabled**

Parameter	Type	Description
detail	String	This parameter is returned only when the tracker status is **error**. It indicates the cause of the abnormal status, and its value can be **bucketPolicyError**, **noBucket**, or **arrears**.
is_obs_created	Boolean	Whether an OBS bucket is created. If the value is **true**, an OBS bucket will be created to store trace files. If the value is **false**, trace files will be stored in an existing OBS bucket. A bucket name contains 3 to 64 characters, including digits, letters, and a hyphen (-) or a period (.).
is_support_trace_files_encryption	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter must be used with **kms_id**.
kms_id	String	Key ID used for transferring and encrypting trace files. This key ID is obtained from Key Management Service (KMS). This parameter is mandatory when **is_support_trace_files_encryption** is set to **true**.
group_id	String	Unique ID of a log group.
stream_id	String	Unique ID of a log stream.
lts	**Lts** object	Trace analysis.
is_support_validate	Boolean	Whether to enable trace file verification.

Table 7-32 Lts

Parameter	Type	Description
is_lts_enabled	Boolean	Whether the LTS search function is enabled.
log_group_name	String	Name of the log group that CTS creates in LTS.
log_topic_name	String	Name of the log topic that CTS creates in LTS.

Example Request

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

Example Response

Status code: 200

The request is successfully sent.

```
{
  "id" : "2e6fa9b8-8c6e-456d-b5d3-77be972d220b",
  "create_time" : 1587958482923,
  "domain_id" : "aexxxxxxxx4d4fb4bexxxxxxx791fbf",
  "is_support_trace_files_encryption" : true,
  "kms_id" : "13a4207c-7abe-4b68-8510-16b84c3b5504",
  "bucket_name" : "obs-f1da",
  "file_prefix_name" : "yO8Q",
  "project_id" : "bb1xxxxxxxxe4f498cbxxxxxxxx35634",
  "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"
}
```

Status Codes

Status Code	Description
200	The request is successfully sent.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
500	Failed to complete the request because of an internal service error.

Error Codes

For details, see **Error Codes**.

7.3.3 Deleting a Tracker

Function

This API is used to delete a tracker from CTS. Deleting a tracker has no impact on the operation records that have been generated. When you enable CTS again, you can still view those traces.

URI

DELETE /v1.0/{project_id}/tracker

Table 7-33 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID.

Table 7-34 Query parameter

Parameter	Mandatory	Type	Description
tracker_name	Yes	String	Tracker name. If this parameter is not specified, all trackers of a tenant will be deleted. Currently, only one tracker is allowed for each tenant. The tracker name is **system**.

Request Parameters

None

Response Parameters

None

Example Request

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

Example Response

None

Status Codes

Status Code	Description
204	The deletion is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The server failed to find the requested resource or some trackers failed to be deleted.

Status Code	Description
500	The request failed to be executed or some trackers failed to be deleted.

Error Codes

For details, see **Error Codes**.

7.3.4 Modifying a Tracker

Function

This API is used to modify configurations of a tracker, including trace transfer to OBS buckets, key event notifications, trace file encryption, trace search and analysis using LTS, trace file integrity check, and tracker enablement or disablement. Modifying tracker parameters does not affect the existing operation records. After the modification is complete, the new rules are immediately applied to operation recording.

URI

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

Table 7-35 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID.
tracker_name	Yes	String	Tracker name. Currently, only one tracker is allowed for each tenant. The tracker name is **system**.

Request Parameters

Table 7-36 Request body parameters

Parameter	Mandatory	Type	Description
bucket_name	No	String	OBS bucket name. The value contains 3 to 63 characters and must start with a digit or lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
file_prefix_name	No	String	Prefix of trace files that need to be stored in OBS buckets. The value can contain 0 to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
status	No	String	Tracker status. The value can be **enabled** or **disabled**. If you change the value to **disabled**, the tracker stops recording traces. Value: ● **enabled** ● **disabled**
is_obs_created	No	Boolean	Whether an OBS bucket is created. If the value is **true**, an OBS bucket will be created to store trace files. If the value is **false**, trace files will be stored in an existing OBS bucket. A bucket name contains 3 to 64 characters, including digits, letters, and a hyphen (-) or a period (.).
is_support_trace_files_encryption	No	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter must be used with **kms_id**.

Parameter	Mandatory	Type	Description
kms_id	No	String	Key ID used for transferring and encrypting trace files. This key ID is obtained from Key Management Service (KMS). This parameter is mandatory when **is_support_trace_files_encryption** is set to **true**.
lts	No	**Lts** object	Trace analysis.
log_file_validate	No	**LogFileValidate** object	File verification.

Table 7-37 Lts

Parameter	Mandatory	Type	Description
is_lts_enabled	Yes	Boolean	Whether the LTS search function is enabled.
log_group_name	Yes	String	Name of the log group that CTS creates in LTS.
log_topic_name	Yes	String	Name of the log topic that CTS creates in LTS.

Table 7-38 LogFileValidate

Parameter	Mandatory	Type	Description
is_support_validate	Yes	Boolean	Whether to enable trace file verification.

Response Parameters

None

Example Request

```
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"
}
```

Example Response

None

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	The request is rejected due to authentication failure.
403	The server understood the request but refused to authorize it.
404	The server failed to find the requested resource.
500	The request failed to be executed or some trackers failed to be deleted.

Error Codes

For details, see **Error Codes**.

8 Permissions Policies and Supported Actions

This section describes fine-grained permissions management for your CTS. If your account does not require individual IAM users, you can skip this section.

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

You can grant users permissions by using **roles** and **policies**. Roles are a type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

NOTE

Policy-based authorization is useful if you want to allow or deny the access to an API.

An account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user wants to query traces using an API, the user must have been granted permissions that allow the **cts:trace:list** action.

Supported Actions

CTS provides system-defined policies that can be directly used in IAM. As the enterprise administrator, you can also create custom policies and use them to supplement system-defined policies, implementing more refined access control. Operations supported by policies are specific to APIs. The following are common concepts related to policies:

- Permissions: Defined by actions in a custom policy.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.

- Actions: Specific operations that are allowed or denied.
- Related actions: Actions on which a specific action depends to take effect. When assigning permissions for the action to a user, you also need to assign permissions for the related actions.
- IAM or enterprise projects: Type of projects for which an action will take effect. Policies that contain actions for both IAM and enterprise projects can be used and take effect for both IAM and Enterprise Management. Policies that only contain actions for IAM projects can be used and only take effect for IAM.

NOTE

The check mark (√) and cross symbol (x) indicate that an action takes effect or does not take effect for the corresponding type of projects.

Permission	API	Action	Related Action	IAM Project	Enterprise Project
Querying a trace list	GET /v3/{project_id}/traces	cts:trace:list	-	√	x
Querying a trace list	GET /v2.0/{project_id}/{tracker_name}/trace	cts:trace:list	-	√	x
Querying a trace list	GET /v1.0/{project_id}/{tracker_name}/trace	cts:trace:list	-	√	x
Querying a tracker	GET /v3/{project_id}/trackers	cts:tracker:list	obs:bucket:GetBucketAcl obs:bucket:ListAllMyBuckets	√	x
Querying a tracker	GET /v1.0/{project_id}/tracker	cts:tracker:list	obs:bucket:GetBucketAcl obs:bucket:ListAllMyBuckets	√	x

Permission	API	Action	Related Action	IAM Project	Enterprise Project
Creating a tracker	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:PutBucketAclkms:cmk:list	√	x
Creating a tracker	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:PutBucketAclkms:cmk:list	√	x

Permission	API	Action	Related Action	IAM Project	Enterprise Project
Modifying a tracker	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
Modifying a tracker	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
Deleting a tracker	DELETE /v3/{project_id}/trackers	cts:tracker:delete	-	√	x

Permission	API	Action	Related Action	IAM Project	Enterprise Project
Deleting a tracker	DELETE /v1.0/{project_id}/tracker	cts:tracker:delete	-	√	x
Querying the tracker quota	GET /v3/{project_id}/quotas	cts:quota:get	-	√	x

9 Appendix

9.1 Error Codes

If an error code starting with **APIGW** is returned after you call an API, rectify the fault according to **APIG Error Codes**.

Status Code	Error Code	Error Message	Description	Solution
400	CTS.0001	The IAM or OBS service is abnormal.	The IAM or OBS service is abnormal.	Contact technical support.
400	CTS.0003	The message body is empty or invalid.	The message body is empty or invalid.	Verify the body content and format.
400	CTS.0200	The number of trackers has reached the upper limit.	The number of trackers has reached the upper limit.	Delete or modify unnecessary trackers.
400	CTS.0201	A management tracker has been created.	A management tracker has been created.	Check whether a management tracker is already available.
400	CTS.0202	The value of the tracker_type parameter is incorrect.	The value of **tracker_type** is invalid.	Change its value to **system** or **data**.

Status Code	Error Code	Error Message	Description	Solution
400	CTS.0203	The value of tracker_name parameter is in an incorrect format.	The value of **tracker_name** is invalid.	Modify its value by referring to the parameter descriptions.
400	CTS.0204	The tracker_name parameter of a management tracker can only be set to system.	The **tracker_name** parameter of a management tracker can only be set to **system**.	Modify its value by referring to the parameter descriptions.
400	CTS.0205	The status parameter can only be set to enabled or disabled.	The **status** parameter can only be set to **enabled** or **disabled**.	Change its value to **enabled** or **disabled**.
400	CTS.0206	The data_bucket parameter cannot be included in the message body for a management tracker.	The **data_bucket** parameter cannot be included in the message body for a management tracker.	Delete the **data_bucket** parameter.
400	CTS.0207	The tracker_name parameter in the message body cannot be set to system for a data tracker.	The **tracker_name** parameter in the message body cannot be set to **system** for a data tracker.	Change the value of **tracker_name** to a value other than **system**.
400	CTS.0208	The tracker already exists.	The tracker already exists.	Check whether the tracker already exists.

Status Code	Error Code	Error Message	Description	Solution
400	CTS.0209	A type of operations on an OBS bucket can be tracked by only one tracker.	A type of operations on an OBS bucket can be tracked by only one tracker.	Change the tracker configurations.
400	CTS.0210	The OBS bucket to track cannot be empty.	The OBS bucket to be tracked cannot be empty.	Select another bucket or ensure that the bucket is not empty.
400	CTS.0211	The tracked OBS bucket does not exist.	The OBS bucket to be tracked does not exist.	Check whether **bucket_name** is correctly set.
400	CTS.0212	The tracked OBS bucket cannot be modified.	The tracked OBS bucket cannot be modified.	Withdraw the changes on the OBS bucket.
400	CTS.0213	The OBS bucket used for trace transfer cannot be a tracked OBS bucket.	The OBS bucket used for trace transfer cannot be a tracked OBS bucket.	Select another OBS bucket for trace transfer.
400	CTS.0215	The OBS bucket already exists.	The OBS bucket already exists.	Change the value of **bucket_name**.
400	CTS.0216	Failed to create a bucket.	Failed to create a bucket.	Contact technical support.
400	CTS.0217	Failed to set a lifecycle rule for the OBS bucket.	Failed to set a lifecycle rule for the OBS bucket.	Contact technical support.
400	CTS.0218	The value of file_prefix_name is in an incorrect format.	The value of **file_prefix_name** is invalid.	Modify its value by referring to the parameter descriptions.

Status Code	Error Code	Error Message	Description	Solution
400	CTS.0219	The operation type cannot be empty.	The operation type cannot be empty.	Select at least one operation type to track.
400	CTS.0220	KMS is not supported.	KMS is not supported.	Contact technical support.
400	CTS.0221	The KMS ID is empty.	The KMS ID is empty.	Check whether the KMS ID is correct.
400	CTS.0222	KMS verification failed.	KMS verification failed.	Check whether the KMS ID is correct.
400	CTS.0225	Only WRITE and/or READ operations on the OBS bucket can be tracked.	The bucket operation must be write, read, or read/write.	Check whether the input parameters are correctly set.
400	CTS.0228	The CTS service is not trusted.	CTS is not trusted.	Enable CTS as a trusted service on the Organizations console.
400	CTS.0229	The organization tracker already exists.	The organization tracker already exists.	Disable the enabled organization tracker first.
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.	Invalid bucket name. A bucket name must contain 3 to 63 characters and start with a digit or a lowercase letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.	Check whether the bucket name is correct.
400	CTS.0300	Query failed.	Query failed.	Try again later or contact technical support.

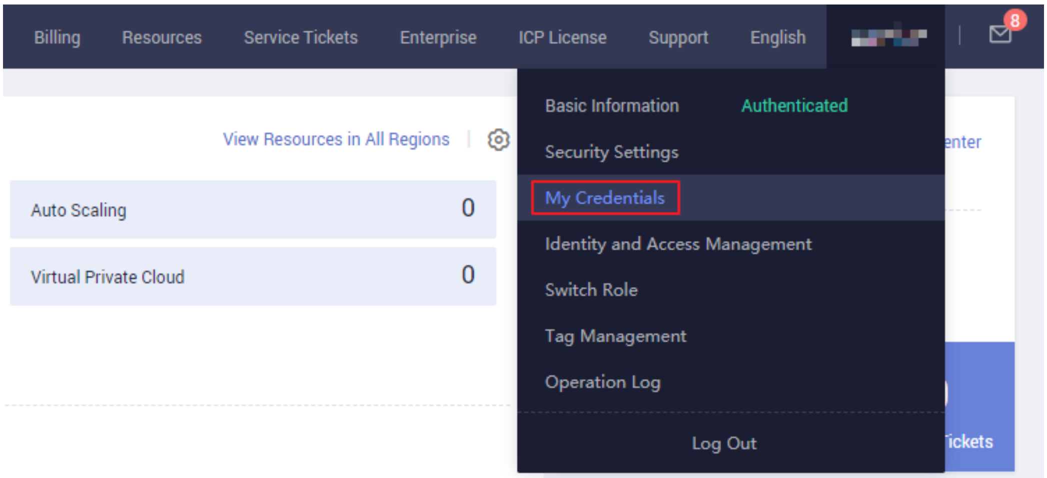
Status Code	Error Code	Error Message	Description	Solution
403	CTS.0002	Authentication failed or you do not have the permissions required.	Authentication failed or you do not have the permissions required.	Check your permissions.
403	CTS.0013	No permission, Please check roles.	You do not have the corresponding operation permission.	Configure the permission.
404	CTS.0100	API version query is not supported in CTS.	API version query is not supported in CTS.	Contact technical support.
404	CTS.0214	The tracker does not exist.	The tracker does not exist.	Check whether the tracker has been deleted.
500	CTS.0004	Failed to write data.	Failed to write data.	Contact technical support.
500	CTS.0005	Failed to read data.	Failed to read data.	Contact technical support.

9.2 Obtaining an Account ID and Project ID

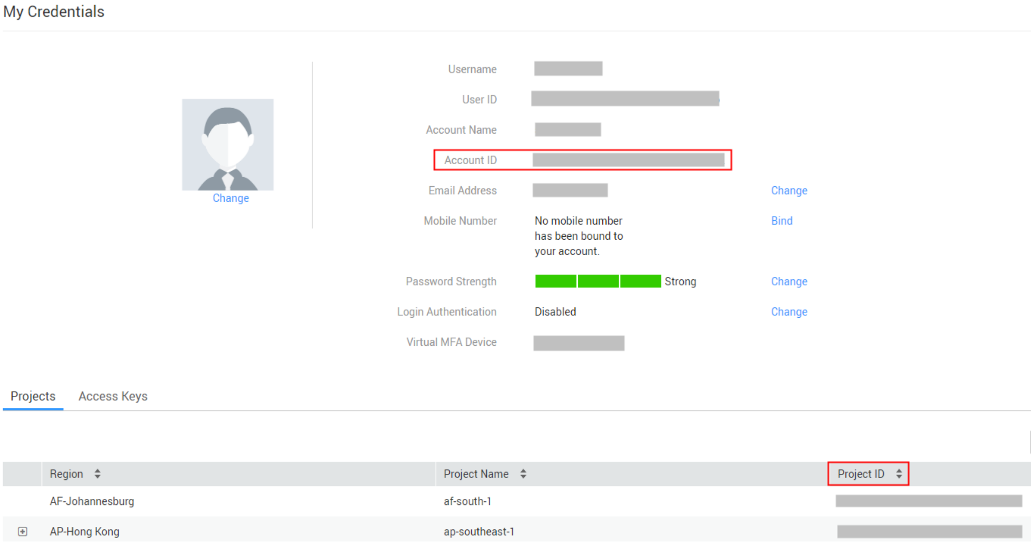
Obtaining Account and Project IDs from the Console

Account ID (**domain-id**) and project ID are required for some URLs when an API is called. You can perform the following operations to obtain these IDs:

1. Log in to the management console. Hover the mouse pointer over the username and choose **My Credentials** from the drop-down list.



2. On the **My Credentials** page, view the account and project IDs.


My Credentials

Change

Username	[redacted]	
User ID	[redacted]	
Account Name	[redacted]	
Account ID	[redacted]	
Email Address	[redacted]	Change
Mobile Number	No mobile number has been bound to your account.	Bind
Password Strength	🟩🟩🟩 Strong	Change
Login Authentication	Disabled	Change
Virtual MFA Device	[redacted]	

Projects Access Keys

	Region	Project Name	Project ID
	AF-Johannesburg	af-south-1	[redacted]
⊞	AP-Hong Kong	ap-southeast-1	[redacted]

If there are multiple projects in one region, expand **Region** and view sub-project IDs from the **Project ID** column.

Obtaining Project IDs by Calling an API

The API for obtaining a project ID is **GET https://*{Endpoint}*/v3/projects**. *{Endpoint}* indicates the endpoint of IAM.

In the following example, **id** indicates a project 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 Change History

Released On	Description
2024-03-28	This issue is the twenty-fourth official release. Incorporated the following changes: ● Added the description of field **agency_name** in section "Creating a Key Event Notification." ● Added the description of field **agency_name** in section "Modifying a Key Event Notification." ● Added the description of fields **access_key_id**, **enterprise_project_id**, **resource_account_id**, and **user** in section "Querying a Trace List." ● Added the description of field **agency_name** in section "Creating a Tracker." ● Added the description of field **agency_name** in section "Modifying a Tracker." ● Added the description of field **agency_name** in section "Querying a Tracker."
2023-9-21	This issue is the twenty-third official release. Incorporated the following changes: Added the description of parameters **is_organization_tracker** and **management_event_selector** in the tracker management APIs.

Released On	Description
2023-09-12	This is the twenty-second official release. Incorporated the following changes: ● Added the following V3 APIs: – Querying All Operations on a Cloud Service – Querying the List of Operators of Audit Logs – Checking Whether Data Can Be Transferred to the Configured OBS Bucket – Querying the Resources Involved in the Traces – Adding CTS Resource Tags in Batches – Deleting CTS Resource Tags in Batches
2023-03-31	This issue is the twenty-first official release. Incorporated the following changes: Added the description of **compress_type** and **is_sort_by_service** to "Creating a Tracker".
2021-10-15	This issue is the twentieth official release. Incorporated the following changes: ● Added the following V3 APIs: – Creating a key event notification – Modifying a key event notification – Querying a key event notification – Deleting a key event notification
2020-08-18	This issue is the nineteenth official release. Incorporated the following changes: ● Renamed section "API (V1) as "Out-of-Date APIs". ● Added the following sections: – API Calling – Endpoints – Constraints – Concepts – API Versions

Released On	Description
2020-06-30	This issue is the eighteenth official release. Incorporated the following changes: ● Added the following V3 APIs: – Querying traces – Creating a tracker – Modifying a tracker – Querying a tracker – Deleting a tracker – Querying the tracker quota ● Added the following API calling examples: – Example 1: Creating a management tracker – Example 2: Querying a tracker list – Example 3: Querying management traces
2019-09-03	This issue is the seventeenth official release. Incorporated the following changes: Added section "Querying Traces (v2.0)".
2019-07-24	This issue is the sixteenth official release. Incorporated the following changes: Modified section "Obtaining the Account ID and Project ID."
2019-01-30	This issue is the fifteenth official release. Incorporated the following changes: Modified section "Error Codes" and added error messages.
2018-09-30	This issue is the fourteenth official release. Incorporated the following changes: Added section "API Version Management". API versions can be queried.
2018-01-30	This issue is the twelfth official release. Incorporated the following changes: ● In sections "Creating a Tracker", "Modifying a Tracker", and "Querying a Tracker", added the description of the **All** option for the **Operation Type** field in key event notifications. ● Added error codes cts.0042 to cts.0045 and cts.0048.

Released On	Description
2017-11-25	This issue is the eleventh official release. Incorporated the following changes: ● In sections "Creating a Tracker" and "Modifying a Tracker", added fields to support the verification of trace file integrity. ● In sections "Creating a Tracker" and "Modifying a Tracker", added fields to support notifications of login operations in key event notifications.
2017-10-30	This issue is the tenth official release. Incorporated the following changes: In sections "Creating a Tracker" and "Modifying a Tracker", added fields to support management trace analysis using LTS.
2017-09-30	This issue is the ninth official release. Incorporated the following changes: ● Added the description of the key event notification API. ● In sections "Creating a Tracker", "Modifying a Tracker", and "Querying a Tracker", added descriptions of bucket creation and fields about KMS. ● In section "Error Codes", added the OBS error information.
2017-08-30	This issue is the eighth official release. Incorporated the following changes: In section "Querying Traces (v1.0)", added the descriptions of fields **request_id**, **location_info**, **endpoint**, and **resource_url**.
2017-05-26	This issue is the seventh official release. Incorporated the following changes: In section "Modifying a Tracker", modified the description of API response.
2017-04-28	This issue is the sixth official release. Incorporated the following changes: Added section "Querying Traces (v2.0)".
2017-02-27	This issue is the fifth official release. Incorporated the following changes: ● In section "Querying a Tracker", modified the description of return code 200. ● In section "Querying Traces", added the description of the **user** field

Released On	Description
2017-02-08	This issue is the fourth official release. Incorporated the following changes: ● In section "Modifying a Tracker", added the description of return code 400. ● In section "Querying a Tracker", added the description of return code 400. ● In section "Deleting a Tracker", added the description of return code 400. ● In section "Error Codes", modified the description of error code cts.0005 and cts.0007.
2017-02-03	This issue is the third official release. Incorporated the following changes: ● In section "Creating a Tracker", added the description of return code 400. ● In section "Modifying a Tracker", added the naming rule for **bucket_name** and added the description of **tracker_name** to the URI part. ● In sections "Querying a Tracker" and "Deleting a Tracker", removed the description of **tracker_name** from the Request Parameters part and URI part. ● In section "Error Codes", modified the description of the error code cts.0023. ● In section "Querying Traces", modified descriptions of parameters **trace_id**, **response**, **code**, **message**, **from**, and **to**. Added descriptions of return codes 400 and 404.
2017-01-20	This issue is the second official release. Incorporated the following changes: Added the **servicetype** field to the API used for querying traces.
2016-12-30	This issue is the first official release.