Cloud Trace Service

API Reference

Issue 01

Date 2025-07-18

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

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. For details about all supported operations, see **API Overview**.

Before calling CTS APIs, ensure that you are familiar with CTS concepts and functions. For details, see *CTS Service Overview*.

Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. To obtain the regions and endpoints, contact the enterprise administrator.

Constraints

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

Concepts

- Trackers

  Before using CTS, you need to enable it. A tracker is automatically created when CTS is enabled. The tracker identifies and associates with all cloud services you are using, and records all operations on the services.

  A management tracker and 100 data trackers can be created for a tenant.

- Traces

  Traces are cloud resource operation logs captured and stored by CTS. You can view traces to identify when operations were performed by which users for tracking.

There are two types of traces. Management traces are operation records reported by cloud services, whereas data traces are read/write operation records reported by Object Storage Service (OBS).

- Domain

  An account is created upon successful registration with the cloud system. 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 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 using an account to use cloud services. Each IAM user has their own identity credentials (password and access keys).

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

- Region

  A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.

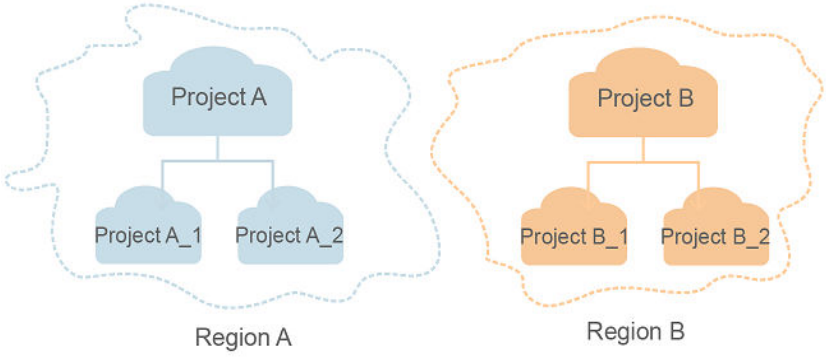
- AZ

  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. An enterprise project can contain resources in multiple regions, and resources can be transferred between enterprise projects.

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

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 the trace list, or creating a tracker.

Table 2-1 lists CTS APIs.

Table 2-1 CTS APIs

Subtype	Description
Trackers	APIs for creating, modifying, querying, and deleting a tracker
Traces	API for querying traces recorded in the last seven days

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 the enterprise administrator.
- **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://{{endpoint}}/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 the project ID by following the instructions in **Obtaining the Account ID and Project ID**.	No	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://{{endpoint}}/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 ***xxxxxxxxxxxxxxxx*** (project name) with the actual values. Obtain a project name from the enterprise administrator.

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. For details, see Obtaining a User Token.

```
POST https://{{endpoint}}/v3/auth/tokens
Content-Type: application/json

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

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:

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

AK/SK-based Authentication

An AK/SK is used to verify the identity of a request sender. In AK/SK-based authentication, a signature needs to be obtained and then added to the request header.

NOTE

- 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.

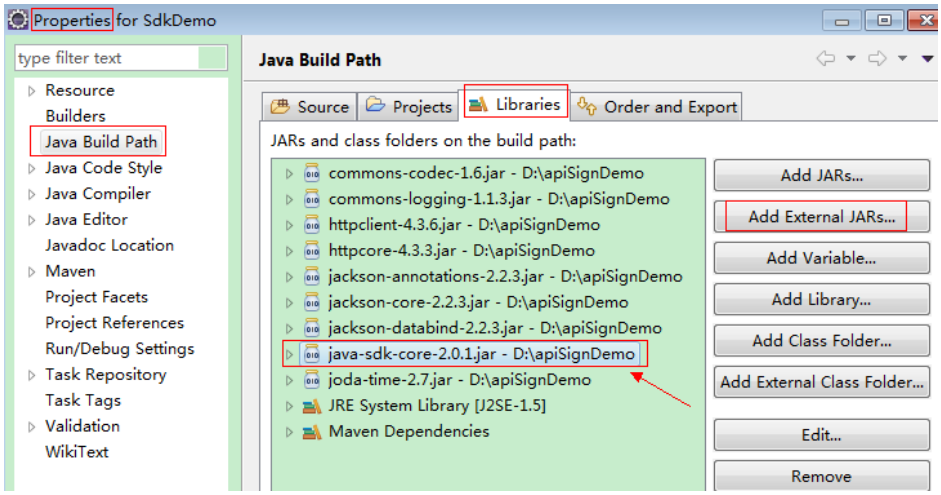
The following demo shows how to sign a request and use an HTTP client to send an HTTPS request.

Download the demo project from **https://github.com/api-gate-way/SdkDemo**.

If you do not need the demo project, download the API Gateway signing tool and reference it.

Obtain the download address of the API Gateway signing tool from the enterprise administrator.

Decompress the downloaded package to obtain a JAR file. Reference the JAR file to the dependency path, as shown below.



Step 1 Generate an AK/SK. If an AK/SK file has already been obtained, skip this step and locate the downloaded AK/SK file. Generally, the file name will be **credentials.csv**.

1. Log in to the console.
2. Hover the mouse pointer over the username and choose **My Credentials** from the drop-down list.
3. In the navigation pane, choose **Access Keys**.
4. Click **Create Access Key**.
5. On the displayed page, enter the login password.
6. Enter the verification code received by email or SMS message.

   NOTE

   For users created in IAM, if no email address or phone number was specified during the user creation, only a login password is required.
7. Click **OK** to download the access key.

   NOTE

   Keep the access key secure.

Step 2 Decompress the demo project.

Step 3 Import the demo project into Eclipse.

Figure 3-1 Selecting Existing Projects into Workspace

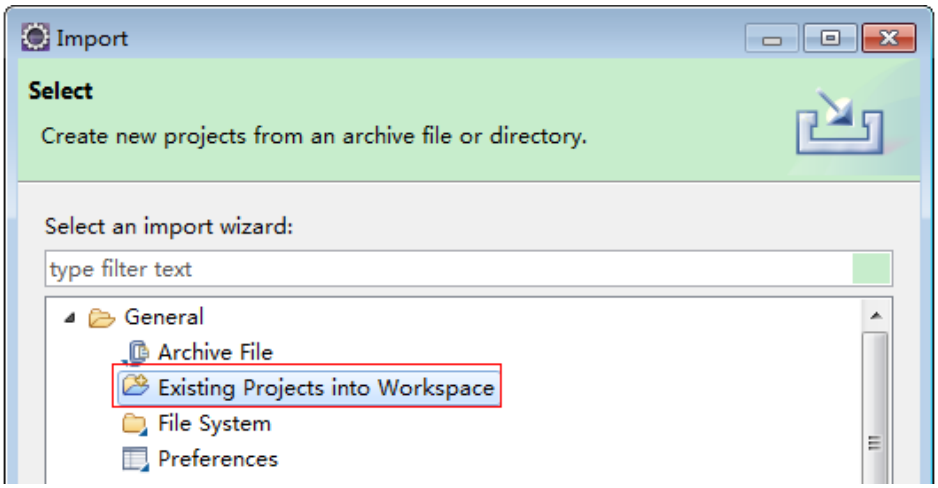


Figure 3-2 Selecting the demo project

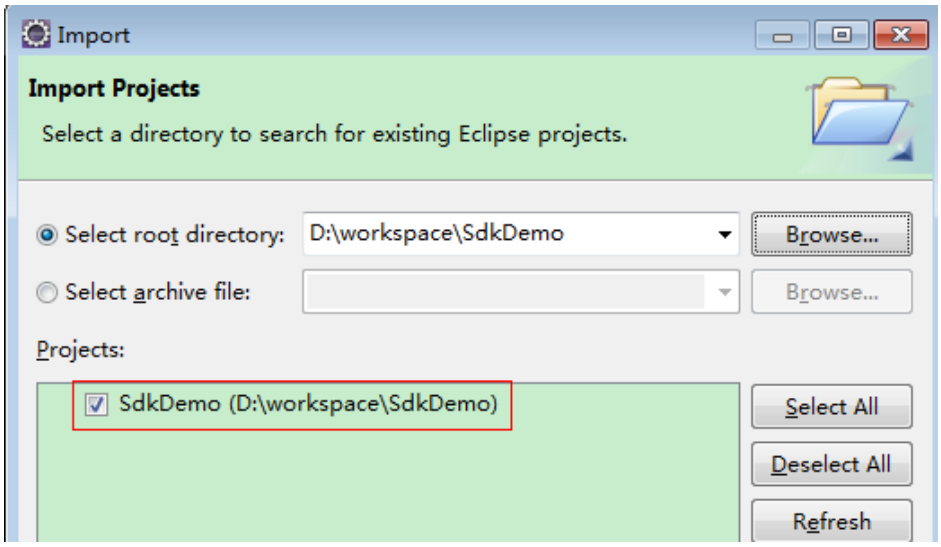
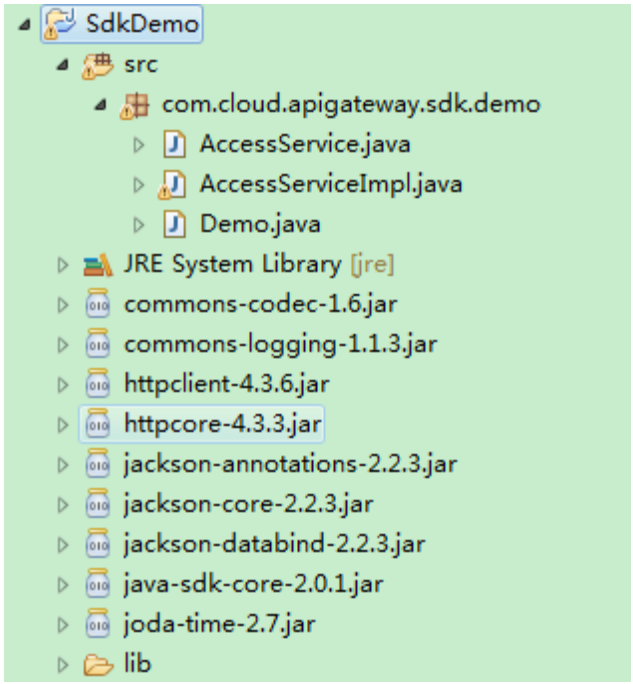


Figure 3-3 Example structure



Step 4 Sign the request.

The request signing method is integrated in the JAR files imported in 3. The request needs to be signed before it is sent. The signature will then be added as part of the HTTP header to the request.

The demo code is classified into three parts:

- **AccessService**: an abstract class that merges the GET, POST, PUT, and DELETE methods into the **access** method.
- **Demo**: an execution entry used to simulate the sending of GET, POST, PUT, and DELETE requests.

- **AccessServiceImpl**: implements the **access** method, which contains the code required for communicating with API Gateway.

1. (Optional) Add request header fields.

   Locate the following rows in the **AccessServiceImpl.java** file, cancel code line shielding, and replace the subproject ID and account ID with the actual ones.

   ```
   //TODO: Add special headers.
   //request.addHeader("X-Project-Id", "xxxxx");
   //request.addHeader("X-Domain-Id", "xxxxx");
   ```

2. Edit the main() method in the **Demo.java** file, and replace the bold text with actual values.

   Replace the bold texts with actual values. If you use other methods, such as POST, PUT, and DELETE, see the corresponding comments.

   Replace **region**, **serviceName**, **ak/sk**, and **url** with the actual values. In the demo, the URL for querying a VPC was used. Replace it with the required URL. To obtain the project ID needed in the URL, see **Obtaining the Account ID and Project ID**. To obtain the endpoint, see **Endpoints**.

   ```
   //TODO: Replace region with the name of the region in which the service to be accessed is located.
    private static final String region = "";

    //TODO: Replace vpc with the name of the service you want to access. For example, ecs, vpc, iam, and elb.
    private static final String serviceName = "";

    public static void main(String[] args) throws UnsupportedEncodingException
    {
    //TODO: Replace the AK and SK with those obtained on the My Credential page.
    //TODO: Directly writing AK/SK in code is risky. For security, encrypt your AK/SK and store them in the configuration file or environment variables.
    //TODO: Before running this example, set environment variables HUAWEICLOUD_SDK_AK and HUAWEICLOUD_SDK_SK.
    String ak = System.getenv("HUAWEICLOUD_SDK_AK");
    String sk = System.getenv("HUAWEICLOUD_SDK_SK");

    //TODO: To specify a project ID (multi-project scenarios), add the X-Project-Id header.
    //TODO: To access a global service, such as IAM, DNS, CDN, and TMS, add the X-Domain-Id header to specify an account ID.
    //TODO: To add a header, find "Add special headers" in the AccessServiceImple.java file.

    //TODO: Test the API
    String url = "https://{Endpoint}/v1/{project_id}/vpcs";
    get(ak, sk, url);

    //TODO: When creating a VPC, replace {project_id} in postUrl with the actual value.
    //String postUrl = "https://serviceEndpoint/v1/{project_id}/cloudservers";
    //String postbody ="{\"vpc\": {\"name\": \"vpc\",\"cidr\": \"192.168.0.0/16\"}}";
    //post(ak, sk, postUrl, postbody);

    //TODO: When querying a VPC, replace {project_id} in url with the actual value.
    //String url = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
    //get(ak, sk, url);

    //TODO: When updating a VPC, replace {project_id} and {vpc_id} in putUrl with the actual values.
    //String putUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
    //String putbody ="{\"vpc\":{\"name\": \"vpc1\",\"cidr\": \"192.168.0.0/16\"}}";
    //put(ak, sk, putUrl, putbody);

    //TODO: When deleting a VPC, replace {project_id} and {vpc_id} in deleteUrl with the actual values.
    //String deleteUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
    //delete(ak, sk, deleteUrl);
    }
   ```

3. Compile the code and call the API.

In the **Package Explorer** area on the left, right-click **Demo.java** and choose **Run AS** > **Java Application** from the shortcut menu to run the demo code.

You can view API call logs on the console.

----End

Token-based Authentication

NOTE

- The validity period of a token is 24 hours. If a token is used for authentication, cache it to prevent frequent API calling.
- Ensure that the token is valid when you use it. Using a token that will soon expire may cause API calling failures.

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 for obtaining a user token. This API is project-level. When calling it, you must 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 **ABCDEFG....**, add **X-Auth-Token: ABCDEFG....** to a request as follows:

```
POST https://{{endpoint}}/v3.0/OS-USER/users
Content-Type: application/json
X-Auth-Token: ABCDEFG....
```

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 3-4 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 3-4 Header fields of the response to the request for obtaining a user token

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 APIs

4.1 Trace Management

4.1.1 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 4-1 Path parameter

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

Table 4-2 Query parameters

Parameter	Mandatory	Type	Description
trace_type	Yes	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 4-3 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 4-4 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** or **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 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.
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	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.
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 resources are operated across tenants.
read_only	boolean	Whether a user request is read-only.
operation_id	String	Operation ID of the trace.

Table 4-5 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 4-6 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 4-7 SessionContext

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

Table 4-8 Attributes

Parameter	Type	Description
created_at	String	Timestamp when a temporary security credential was issued. The value is a 13-digit UTC time in milliseconds.
mfa_authenticated	String	Whether MFA identity authentication has been passed.

Table 4-9 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.

Status code: 400

Table 4-10 Response body parameters

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

Status code: 401

Table 4-11 Response body parameters

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

Parameter	Type	Description
error_msg	String	Error message.

Status code: 403

Table 4-12 Response body parameters

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

Status code: 404

Table 4-13 Response body parameters

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

Status code: 500

Table 4-14 Response body parameters

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

Status code: 503

Table 4-15 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=tracker&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",
    "read_only": false,
    "operation_id": "ListSubscriptions"
  }, {
    "time" : 1472148708232,
    "user" : {
      "name" : "xxx",
      "domain" : {
        "name" : "xxx",
        "id" : "ded649d814464428ba89d04d7955c93e"
      }
    },
    "response" : {
      "code" : "VPC.0514",
      "message" : "Update port fail."
    },
    "code" : 200,
    "service_type" : "VPC",
    "resource_type" : "eip",
    "resource_name" : "192.144.163.1",
    "resource_id" : "d502809d-0d1d-41ce-9690-784282142ccc",
    "trace_name" : "deleteEip",
    "trace_rating" : "warning",
    "trace_type" : "ConsoleAction",
```



```
    "api_version" : "2.0",
    "record_time" : 1481066128032,
    "trace_id" : "e001ccb8-bc09-11e6-b2cc-2640a43cc6e8",
    "read_only": false,
    "operation_id": "ListSubscriptions"
  } ]
}
```

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 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 Code

For details, see **Error Codes**.

4.2 Tracker Management

4.2.1 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 4-16 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 4-17 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.

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 creation. Value: • **cts_admin_trust**

Table 4-18 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 4-19 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**

Table 4-20 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.
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 4-21 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.
management_event_selector	**ManagementEventSelector** object	Management trace selector.
lts	**Lts** object	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**.

Parameter	Type	Description
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**.
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 4-22 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 4-23 Lts

Parameter	Type	Description
is_lts_enabled	Boolean	Whether the LTS search function is enabled.

Parameter	Type	Description
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 4-24 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**.

Table 4-25 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**: read operations of an OBS object; **WRITE**: write operations of an OBS object. Values: • **WRITE** • **READ**

Status code: 400

Table 4-26 Response body parameters

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

Status code: 401

Table 4-27 Response body parameters

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

Status code: 403

Table 4-28 Response body parameters

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

Status code: 404

Table 4-29 Response body parameters

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

Status code: 500

Table 4-30 Response body parameters

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

Status code: 503

Table 4-31 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"
}
```

Status Codes

Status Code	Description
201	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.

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**.

4.2.2 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 4-32 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 4-33 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 4-34 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 4-35 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**

Table 4-36 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.
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: 400

Table 4-37 Response body parameters

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

Status code: 401

Table 4-38 Response body parameters

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

Status code: 403

Table 4-39 Response body parameters

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

Status code: 404

Table 4-40 Response body parameters

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

Status code: 500

Table 4-41 Response body parameters

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

Status code: 503

Table 4-42 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**.

4.2.3 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 4-43 Path parameters

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

Table 4-44 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). Values: • **system** • **data**

Request Parameters

None.

Response Parameters

Status code: 200

Table 4-45 Response body parameters

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

Table 4-46 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**.

Parameter	Type	Description
is_support_validate	Boolean	Whether to enable trace file verification.
lts	**Lts** object	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**.
group_id	String	LTS log group ID.
stream_id	String	LTS log stream ID.

Parameter	Type	Description
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 4-47 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.

Table 4-48 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.

Parameter	Type	Description
bucket_lifecycle	Long	Duration that traces are stored in the OBS bucket. This parameter is valid when **tracker_type** is set to **data**.

Table 4-49 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**: read operations of an OBS object; **WRITE**: write operations of an OBS object. Values: • **WRITE** • **READ**

Table 4-50 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 4-51 Response body parameters

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

Status code: 401

Table 4-52 Response body parameters

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

Status code: 403

Table 4-53 Response body parameters

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

Status code: 500

Table 4-54 Response body parameters

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

Status code: 503

Table 4-55 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"
  } ]
}
```


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

Error Code

For details, see **Error Codes**.

4.2.4 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 4-56 Path parameters

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

Table 4-57 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 4-58 Response body parameters

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

Status code: 401

Table 4-59 Response body parameters

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

Status code: 403

Table 4-60 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 4-61 Response body parameters

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

Status code: 500

Table 4-62 Response body parameters

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

Status code: 503

Table 4-63 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.

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

Error Code

For details, see **Error Codes**.

4.3 Key Event Notification Management

4.3.1 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 4-64 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 4-65 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.

Parameter	Mandatory	Type	Description
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.
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 4-66 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.

Parameter	Mandatory	Type	Description
trace_names	Yes	Array of strings	Trace name.

Table 4-67 NotificationUsers

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

Table 4-68 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: 201

Table 4-69 Response body parameters

Parameter	Type	Description
notification_name	String	Notification name.
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.

Parameter	Type	Description
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.
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 4-70 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.

Parameter	Type	Description
trace_names	Array of strings	Trace name.

Table 4-71 NotificationUsers

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

Table 4-72 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 4-73 Response body parameters

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

Status code: 401

Table 4-74 Response body parameters

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

Status code: 403

Table 4-75 Response body parameters

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

Status code: 404

Table 4-76 Response body parameters

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

Status code: 500

Table 4-77 Response body parameters

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

Status code: 503

Table 4-78 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"
}
```

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**.

4.3.2 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 4-79 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 4-80 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**

Parameter	Mandatory	Type	Description
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**
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 4-81 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 4-82 NotificationUsers

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

Table 4-83 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**

Parameter	Mandatory	Type	Description
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: 200

Table 4-84 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 4-85 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 4-86 NotificationUsers

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

Table 4-87 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 4-88 Response body parameters

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

Status code: 401

Table 4-89 Response body parameters

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

Status code: 403

Table 4-90 Response body parameters

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

Status code: 404

Table 4-91 Response body parameters

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

Status code: 500

Table 4-92 Response body parameters

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

Status code: 503

Table 4-93 Response body parameters

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

Example Request

Modifying 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**.

4.3.3 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 4-94 Path parameters

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

Table 4-95 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 4-96 Response body parameters

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

Status code: 401

Table 4-97 Response body parameters

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

Status code: 403

Table 4-98 Response body parameters

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

Status code: 404

Table 4-99 Response body parameters

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

Status code: 500

Table 4-100 Response body parameters

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

Status code: 503

Table 4-101 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**.

4.3.4 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 4-102 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. Values: • **smn** • **fun**

Table 4-103 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 4-104 Response body parameters

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

Table 4-105 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. 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**. 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**). 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 4-106 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 4-107 NotificationUsers

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

Table 4-108 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 4-109 Response body parameters

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

Status code: 401

Table 4-110 Response body parameters

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

Status code: 403

Table 4-111 Response body parameters

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

Status code: 404

Table 4-112 Response body parameters

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

Status code: 500

Table 4-113 Response body parameters

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

Status code: 503

Table 4-114 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"
  } ]
}
```


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.
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.

Status Code	Description
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**.

4.4 Other APIs

4.4.1 Querying the Tracker Quota of a Tenant

Function

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

API Calling

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/quotas

Table 4-115 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 4-116 Response body parameter

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

Table 4-117 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 4-118 Response body parameters

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

Status code: 401

Table 4-119 Response body parameters

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

Status code: 403

Table 4-120 Response body parameters

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

Status code: 404

Table 4-121 Response body parameters

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

Status code: 500

Table 4-122 Response body parameters

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

Status code: 503

Table 4-123 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 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 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**.

4.5 Tag Management

4.5.1 Adding CTS Resource Tags in Batches

Function

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

API Calling

For details, see **Calling APIs**.

URI

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

Table 4-124 Path parameters

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

Parameter	Mandatory	Type	Description
resource_type	Yes	String	Resource type of CTS. Only **cts-tracker** is supported. Value: • **cts-tracker**
resource_id	Yes	String	Resource ID.

Request Parameters

Table 4-125 Request body parameter

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

Table 4-126 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 4-127 Response body parameters

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

Status code: 403

Table 4-128 Response body parameters

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

Status code: 404

Table 4-129 Response body parameters

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

Status code: 500

Table 4-130 Response body parameters

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

Status code: 503

Table 4-131 Response body parameters

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

Example Request

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**.

4.5.2 Deleting CTS Resource Tags in Batches

Function

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

API Calling

For details, see **Calling APIs**.

URI

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

Table 4-132 Path parameters

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

Request Parameters

Table 4-133 Request body parameter

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

Table 4-134 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 4-135 Response body parameters

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

Status code: 403

Table 4-136 Response body parameters

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

Status code: 404

Table 4-137 Response body parameters

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

Status code: 500

Table 4-138 Response body parameters

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

Status code: 503

Table 4-139 Response body parameters

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

Example Request

Deleting tracker tags in batches

```
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**.

5 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.

Table 5-1 Lifecycle management

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
Creating a key event notification	POST /v3/{project_id}/notifications	cts:notification:create	smn:topic:list	√	x
Modifying a key event notification	PUT /v3/{project_id}/notifications	cts:notification:update	smn:topic:list	√	x
Deleting a key event notification	DELETE /v3/{project_id}/notifications	cts:notification:delete	-	√	x
Querying a key event notification	GET /v3/{project_id}/notifications/{notification_type}	cts:notification:list	-	√	x

6 Appendix

6.1 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.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.

Status Code	Error Code	Error Message	Description	Solution
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.
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.

Status Code	Error Code	Error Message	Description	Solution
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.
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.

Status Code	Error Code	Error Message	Description	Solution
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 contain 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.
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.

Status Code	Error Code	Error Message	Description	Solution
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.

6.2 Obtaining the 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.

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"
  }
}
```