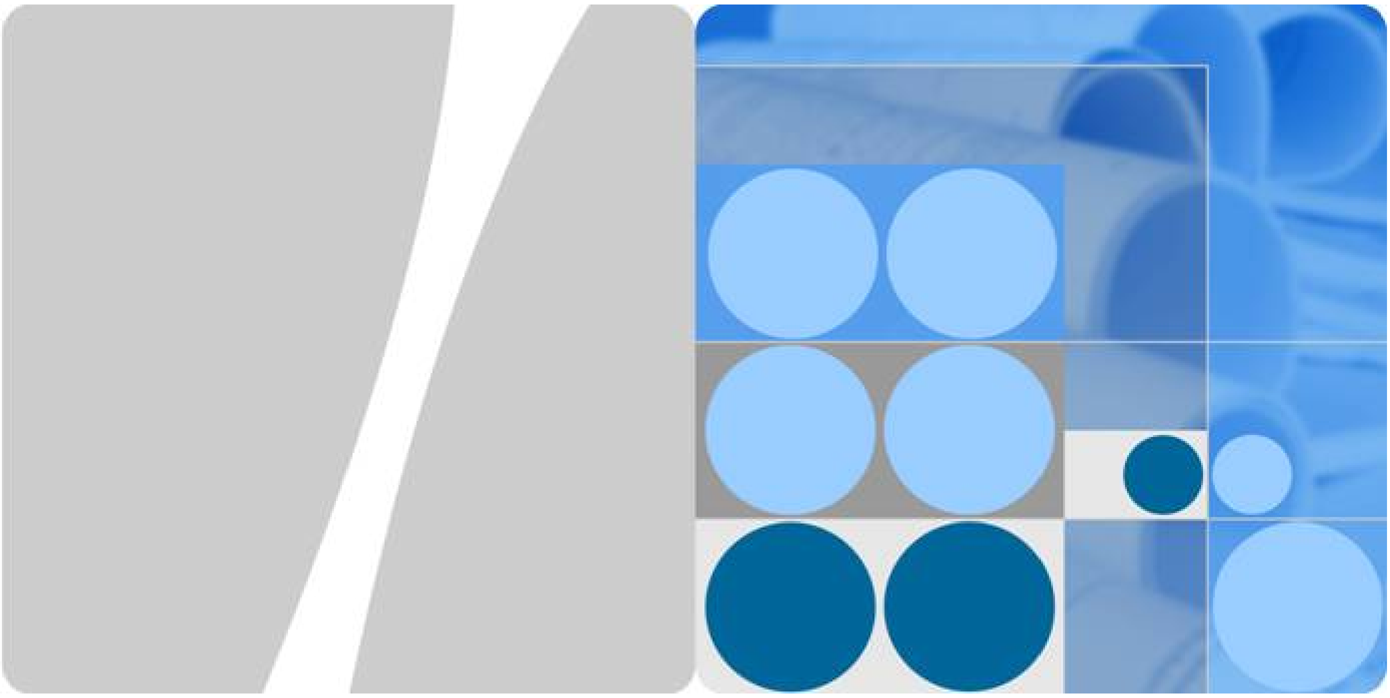


Cloud Trace Service

API Reference (Kuala Lumpur Region)

Date **2022-04-08**

Contents

1 Before You Start

1.1 Overview

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

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

1.2 API Calling

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

1.3 Endpoints

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

1.4 Concepts

- User

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

  API authentication requires information such as the domain name, username, and password.

- 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.

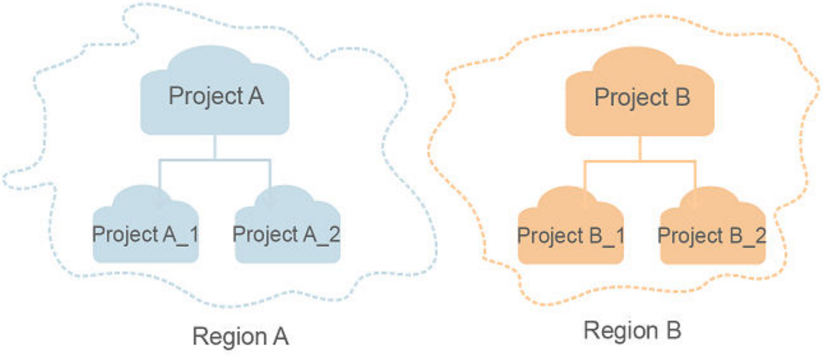
- AZ

  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. Default projects are defined to a group and have physically isolated resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources in the region under their domains. 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



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 1 lists the 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 **Regions and Endpoints**.

  For example, the endpoint of IAM in the **my-kualalumpur-1** region is **iam.my-kualalumpur-1.cloudalphai4.tmone.com.my**.

- **resource-path**:

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

- **query-string**:

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

NOTE

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

Request Methods

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

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

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

```
POST https://iam.my-kualalumpur-1.cloudalphai4.tmone.com.my/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
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

Parameter	Description	Mandatory	Example Value
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: MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

NOTE

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

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

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

```
POST https://iam.my-kualalumpur-1.cloudalphai4.tmone.com.my/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 ***xxxxxxxxxxxxxxxxxx*** (project name) with the actual values. Obtain a project name from **Regions and Endpoints**.

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://iam.my-kualalumpur-1.cloudalphai4.tmone.com.my/v3/auth/tokens
```

```
Content-Type: application/json

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

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

3.2 Authentication

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

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

Token-based Authentication

NOTE

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

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

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

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

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

```
POST https://iam.my-kualalumpur-1.cloudalphai4.tmone.com.my/v3/auth/tokens
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

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

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

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

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

NOTICE

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

3.3 Response

Status Code

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

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

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

Response Header

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

The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

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 Other APIs

4.1.1 Querying the Tracker Quota of a Tenant

Function

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

URI

GET /v3/{project_id}/quotas

Table 4-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see section "Obtaining the Account ID and Project ID" in *Cloud Trace Service API Reference*.

Request Parameters

None

Response Parameters

Status code: 200

Table 4-2 Response body parameters

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

Table 4-3 Quota

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

Status code: 400

Table 4-4 Response body parameters

Parameter	Type	Description
error_code	String	Error code, in the format of **CTS**.*XXX*.
error_msg	String	Error message.

Example Requests

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

Example Responses

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	The server has received the request but encountered an internal error.
503	The requested service is unavailable. The client should not repeat the request without modifications.

Error Codes

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.Operation records (traces) are retained for 7 days and you can check them on the CTS console. To store traces for a longer period, you can transfer them to an Object Storage Service (OBS) bucket in real time.

URI

POST /v3/{project_id}/tracker

Table 4-5 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see section "Obtaining the Account ID and Project ID" in *Cloud Trace Service API Reference*.

Request Parameters

Table 4-6 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**. Enumeration 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 trace analysis is enabled.
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 only when **tracker_type** is set to **system**. It must be used together with **kms_id**.

Parameter	Mandatory	Type	Description
kms_id	No	String	ID of the key used for trace file encryption. The key ID is obtained from Key Management Service (KMS). This parameter is valid only 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**.

Table 4-7 TrackerObsInfo

Parameter	Mandatory	Type	Description
bucket_name	No	String	OBS bucket name. The value contains 3 to 63 characters and must start with a number or lowercase letter. Only lowercase letters, numbers, hyphens (-), and periods (.) are allowed.
file_prefix_name	No	String	File name prefix to mark trace files that need to be stored in an OBS bucket. The value contains 0 to 64 characters. Only letters, numbers, hyphens (-), underscores (_), and periods (.) are allowed.
is_obs_created	No	Boolean	Whether a new OBS bucket is created. When the value is **true**, you can create an OBS bucket to store trace files. When the value is **false**, you can select an existing OBS bucket to store trace files.

Parameter	Mandatory	Type	Description
bucket_lifecycle	No	Integer	Duration that traces are stored in the OBS bucket. This parameter is valid only when **tracker_type** is set to **data**. Enumeration values: - **30** - **60** - **90** - **180** - **1095**

Table 4-8 DataBucket

Parameter	Mandatory	Type	Description
data_bucket_name	No	String	Name of the bucket tracked by a data tracker. - This parameter is mandatory when you enable or disable a data tracker. - 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 you enable or disable a data tracker. - This parameter is unavailable for a management tracker. Enumeration values: - **WRITE** - **READ**

Response Parameters

Status code: 201

Table 4-9 Response body parameters

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

Parameter	Type	Description
data_bucket	**DataBucketQuery** object	Information about the bucket tracked by a data tracker. This parameter is valid only when the value of **tracker_type** is **data**.

Table 4-10 Lts

Parameter	Type	Description
is_lts_enabled	Boolean	Whether traces are synchronized to LTS for trace search and analysis.
log_group_name	String	Name of the log group that CTS creates in LTS.
log_topic_name	String	Name of the log stream that CTS creates in LTS.

Table 4-11 ObsInfo

Parameter	Type	Description
bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a number or lowercase letter. Only lowercase letters, numbers, hyphens (-), and periods (.) are allowed.
file_prefix_name	String	File name prefix to mark trace files that need to be stored in an OBS bucket. The value contains 0 to 64 characters. Only letters, numbers, hyphens (-), underscores (_), and periods (.) are allowed.
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 only when **tracker_type** is set to **data**.

Table 4-12 DataBucketQuery

Parameter	Type	Description
data_bucket_name	String	OBS bucket name. The value contains 3 to 63 characters and must start with a number or lowercase letter. Only lowercase letters, numbers, 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 you enable or disable a data tracker. ● This parameter is unavailable for a management tracker. ● Once a tracker is created, the bucket that it tracks cannot be changed. Enumeration values: ● **WRITE** ● **READ**

Status code: 400

Table 4-13 Response body parameters

Parameter	Type	Description
error_code	String	Error code, in the format of **CTS**.*XXX*.
error_msg	String	Error message.

Example Requests

- Creating a management tracker

```
POST https://{endpoint}/v3/{project_id}/tracker

{
  "tracker_type" : "system",
  "tracker_name" : "system",
  "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",
  "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 Responses

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",
  "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"
  },
  "log_file_validate" : {
    "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.

Status Code	Description
500	The server has received the request but encountered an internal error.
503	The requested service is unavailable. The client should not repeat the request without modifications.

Error Codes

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, management trace retrieval using Log Tank Service (LTS), trace file integrity check, and tracker enablement or disablement. Modifying tracker parameters does not affect the collected traces. After the modification is complete, the new rules are immediately applied to operation recording.

URI

PUT /v3/{project_id}/tracker

Table 4-14 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see section "Obtaining the Account ID and Project ID" in *Cloud Trace Service API Reference*.

Request Parameters

Table 4-15 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**. Enumeration 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. Enumeration values: - **enabled** - **disabled**
is_lts_enabled	No	Boolean	Whether trace analysis is enabled.
obs_info	No	**TrackerObsInfo** object	Configurations of an OBS bucket to which traces are transferred.

Parameter	Mandatory	Type	Description
is_support_trace_files_encryption	No	Boolean	Whether trace files are encrypted during transfer to an OBS bucket. This parameter is valid only when **tracker_type** is set to **system**. It must be used together with **kms_id**.
kms_id	No	String	ID of the key used for trace file encryption. The key ID is obtained from Key Management Service (KMS). This parameter is valid only 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	Configurations of a tracked OBS bucket. This parameter is valid only when **tracker_type** is set to **data**.

Table 4-16 TrackerObsInfo

Parameter	Mandatory	Type	Description
bucket_name	No	String	OBS bucket name. The value contains 3 to 63 characters and must start with a number or lowercase letter. Only lowercase letters, numbers, hyphens (-), and periods (.) are allowed.
file_prefix_name	No	String	File name prefix to mark trace files that need to be stored in an OBS bucket. The value contains 0 to 64 characters. Only letters, numbers, hyphens (-), underscores (_), and periods (.) are allowed.

Parameter	Mandatory	Type	Description
is_obs_created	No	Boolean	Whether a new OBS bucket is created. When the value is **true**, you can create an OBS bucket to store trace files. When the value is **false**, you can select an existing OBS bucket to store trace files.
bucket_lifecycle	No	Integer	Duration that traces are stored in the OBS bucket. This parameter is valid only when **tracker_type** is set to **data**. Enumeration values: • **30** • **60** • **90** • **180** • **1095**

Table 4-17 DataBucket

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

Parameter	Mandatory	Type	Description
data_event	No	Array of strings	Type of operations tracked by a data tracker. ● This parameter is mandatory when you enable or disable a data tracker. ● This parameter is unavailable for a management tracker. Enumeration values: ● **WRITE** ● **READ**

Response Parameters

Status code: 400

Table 4-18 Response body parameters

Parameter	Type	Description
error_code	String	Error code, in the format of **CTS**.*XXX*.
error_msg	String	Error message.

Example Requests

- Modifying a management tracker

```
PUT https://{endpoint}/v3/{project_id}/tracker

{
  "tracker_type" : "system",
  "tracker_name" : "system",
  "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",
  "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 Responses

None

Status Codes

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

Error Codes

See **Error Codes**.

4.2.3 Querying a Tracker

Function

This API is used to query tracker details, including the tracker name, name of OBS buckets for storing traces, and file name prefix of the traces files stored in OBS buckets.

URI

GET /v3/{project_id}/trackers

Table 4-19 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see section "Obtaining the Account ID and Project ID" in *Cloud Trace Service API Reference*.

Table 4-20 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). Enumeration values: ● **system** ● **data**

Request Parameters

None

Response Parameters

Status code: 200

Table 4-21 Response body parameters

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

Table 4-22 TrackerResponseBody

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

Parameter	Type	Description
data_bucket	**DataBucketQuery** object	Information about the bucket tracked by a data tracker. This parameter is valid only when the value of **tracker_type** is **data**.

Table 4-23 Lts

Parameter	Type	Description
is_lts_enabled	Boolean	Whether traces are synchronized to LTS for trace search and analysis.
log_group_name	String	Name of the log group that CTS creates in LTS.
log_topic_name	String	Name of the log stream 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 number or lowercase letter. Only lowercase letters, numbers, hyphens (-), and periods (.) are allowed.
file_prefix_name	String	File name prefix to mark trace files that need to be stored in an OBS bucket. The value contains 0 to 64 characters. Only letters, numbers, hyphens (-), underscores (_), and periods (.) are allowed.
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 only 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 number or lowercase letter. Only lowercase letters, numbers, 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 you enable or disable a data tracker. ● This parameter is unavailable for a management tracker. ● Once a tracker is created, the bucket that it tracks cannot be changed. Enumeration values: ● **WRITE** ● **READ**

Status code: 400

Table 4-26 Response body parameters

Parameter	Type	Description
error_code	String	Error code, in the format of **CTS**.*XXX*.
error_msg	String	Error message.

Example Requests

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

Example Responses

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",
    "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	The server has received the request but encountered an internal error.

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

Error Codes

See **Error Codes**.

4.2.4 Deleting a Tracker

Function

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

URI

DELETE /v3/{project_id}/trackers

Table 4-27 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see section "Obtaining the Account ID and Project ID" in *Cloud Trace Service API Reference*.

Table 4-28 Query Parameters

Parameter	Mandatory	Type	Description
tracker_name	No	String	Tracker name. If this parameter is not specified, all data trackers of a tenant will be deleted.
tracker_type	No	String	Tracker type. Only data trackers can be deleted. The default value is **data**. Enumeration values: ● **data**

Request Parameters

None

Response Parameters

Status code: 400

Table 4-29 Response body parameters

Parameter	Type	Description
error_code	String	Error code, in the format of **CTS**.*XXX*.
error_msg	String	Error message.

Example Requests

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

Example Responses

None

Status Codes

Status Code	Description
204	The tracker is deleted.
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 server has received the request but encountered an internal error, or some trackers failed to be deleted.
503	The requested service is unavailable. The client should not repeat the request without modifications.

Error Codes

See **Error Codes**.

4.3 Trace Management

4.3.1 Querying a Trace List

Function

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

URI

GET /v3/{project_id}/traces

Table 4-30 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see section "Obtaining the Account ID and Project ID" in *Cloud Trace Service API Reference*.

Table 4-31 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**. Enumeration values: ● **system** ● **data**
limit	No	Integer	Number of traces to query. The default value is 10 and the maximum value is 200.
from	No	Long	UTC millisecond timestamp of the query start time. The value contains 13 numbers 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.

Parameter	Mandatory	Type	Description
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 numbers 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 value of this parameter is **system**. 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.
service_type	No	String	Cloud service whose traces are to be queried. The value must be the abbreviation 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 have been connected with CTS, see section "Supported Services and Operation Lists" in the *Cloud Trace Service User Guide*.
user	No	String	Name of a user whose traces are to be queried. This parameter is valid only when **trace_type** is set to **system**.

Parameter	Mandatory	Type	Description
resource_id	No	String	Cloud resource 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**.
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.
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**. Enumeration values: ● **normal** ● **warning** ● **incident**

Request Parameters

None

Response Parameters

Status code: 200

Table 4-32 Response body parameters

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

Table 4-33 Traces

Parameter	Type	Description
resource_id	String	ID of the cloud resource on which the recorded operation was performed.
trace_name	String	Trace name. The value contains 1 to 64 characters and must start with a letter. Only letters, numbers, hyphens (-), underscores (_), and periods (.) are allowed.
trace_rating	String	Trace status. The value can be **normal**, **warning**, or **incident**. Enumeration 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 body of the recorded operation.
response	String	Response body of the recorded operation.
code	String	Returned HTTP status code of the recorded operation.
api_version	String	Version of the API called in the trace.
message	String	Remarks added by other cloud services to the 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.

Parameter	Type	Description
user	**UserInfo** object	Information of the user who performed the operation that triggered the trace.
service_type	String	Cloud service on which the recorded operation was performed. 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 recorded operation was performed.
source_ip	String	IP address of the tenant who performed the operation that triggered the trace.
resource_name	String	Name of the resource on which the recorded operation was performed.
request_id	String	ID of the request of the recorded operation.
location_info	String	Information required for fault locating after a request error occurred.
endpoint	String	Endpoint in the details page URL of the cloud resource on which the recorded operation was performed.
resource_url	String	Details page URL (excluding the endpoint) of the cloud resource on which the recorded operation was performed.

Table 4-34 UserInfo

Parameter	Type	Description
id	String	Account ID. For details, see section "Obtaining the Account ID and Project ID" in *Cloud Trace Service API Reference*.
name	String	Account name.
domain	**BaseUser** object	Domain information of the user who performed the operation that triggered the trace.

Table 4-35 BaseUser

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

Table 4-36 MetaData

Parameter	Type	Description
count	Integer	Number of returned traces.
marker	String	ID of the last trace returned. 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 4-37 Response body parameters

Parameter	Type	Description
error_code	String	Error code, in the format of **CTS**.*XXX*.
error_msg	String	Error message.

Example Requests

- Querying management traces

```
GET https://{endpoint}/v3/{project_id}/traces?
limit=11&to=1479095278000&from=1478490478000&trace_name=createTracker&resource_type=tracke
r&service_type=CTS&trace_type=system
```

- Querying data traces

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

Example Responses

Status code: 200

The request is successful.

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

Status Codes

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

Status Code	Description
500	The server has received the request but encountered an internal error.
503	The requested service is unavailable. The client should not repeat the request without modifications.

Error Codes

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 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 metrics using an API, the user must have been granted permissions that allow the **aom:metric:get** action.

Supported Actions

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

- Permissions: Defined by actions in a custom policy.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.
- Actions: Specific operations that are allowed or denied.

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

NOTE

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

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

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
Deleting a tracker	DELETE /v3/{project_id}/trackers	cts:tracker:delete	-	√	x
Querying the tracker quota	GET /v3/{project_id}/quotas	cts:quota:get	-	√	x

6 Appendix

6.1 Error Codes

Status Code	Error Codes	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.	Check the content and format of the message body.
400	CTS.0200	The number of trackers has reached the upper limit.	The number of trackers has reached the upper limit.	Delete or modify trackers no longer needed.
400	CTS.0201	A management tracker has been created.	A management tracker has been created.	Check whether a management tracker is already available.
400	CTS.0202	The value of the tracker_type parameter is incorrect.	The value of the tracker_type parameter is incorrect.	Change its value to system or data.

Status Code	Error Codes	Error Message	Description	Solution
400	CTS.0203	The value of tracker_name parameter is in an incorrect format.	The value of tracker_name parameter is in an incorrect format.	Modify its value. Ensure that the tracker name is a string of 1 to 32 characters and does not start with an underscore (_) or hyphen (-)
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.	Change the value of tracker_name to system.
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.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 Codes	Error Message	Description	Solution
400	CTS.0210	The OBS bucket to track cannot be empty.	The OBS bucket to track 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 tracked OBS bucket does not exist.	Check whether the 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 in an incorrect format.	Modify its value. Ensure that the file prefix name is a string of 0 to 64 characters and contains only letters, digits, underscores (_), hyphens (-), or periods (.)
400	CTS.0219	The operation type cannot be empty.	The operation type cannot be empty.	Select at least one operation type to track.

Status Code	Error Codes	Error Message	Description	Solution
400	CTS.0220	KMS is not supported.	KMS is not supported.	Contact technical support.
400	CTS.0221	The KMS ID is empty.	The KMS ID is empty.	Check whether the KMS ID is correct.
400	CTS.0222	KMS verification failed.	KMS verification failed.	Check whether the KMS ID is correct.
400	CTS.0225	Only WRITE and/or READ operations on the OBS bucket can be tracked.	Only WRITE and/or READ operations on the OBS bucket can be tracked.	Check whether the input parameters are correctly set.
400	CTS.0231	Invalid bucket name. A bucket name must be a string of 3 to 63 characters, including only lowercase letters, digits, hyphens (-), or periods (.). It must start with a digit or a lowercase letter.	Invalid bucket name. A bucket name must be a string of 3 to 63 characters, including only lowercase letters, digits, hyphens (-), or periods (.). It must start with a digit or a lowercase letter.	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.
403	CTS.0208	The tracker already exists.	The tracker already exists.	Check whether the tracker exists.
404	CTS.0100	API version query is not supported in CTS.	API version query is not supported in CTS.	Contact technical support.

Status Code	Error Codes	Error Message	Description	Solution
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"
    }
}
```