Cloud Stream Service

API Reference

Issue 1.0

Date 2020-03-16

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks 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 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 Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Contents

1 Before You Start

1.1 Overview

Welcome to Cloud Stream Service API Reference. Cloud Stream Service (CS) is a real-time big data stream analysis service running on the public cloud. Computing clusters are fully managed by CS, allowing you to focus on Stream SQL services. CS is fully compatible with Apache Flink 1.5.3 and Apache Spark 2.2.1 APIs.

Promoted in the IT field, CS is a distributed real-time stream computing system featuring low latency (millisecond-level latency), high throughput, and high reliability. Powered on Flink, CS integrates enhanced features and security, and supports both stream processing and batch processing methods. It provides mandatory Stream SQL features for data processing, and will add algorithms of machine learning and graph computing to Stream SQL in the future.

This document describes how to use application programming interfaces (APIs) to perform operations on CS, such as creating, deleting, or deleting CS jobs, clusters, or templates. For details about all supported operations, see **API Overview**.

If you plan to access CS through an API, ensure that you are familiar with CS concepts. For details, see **Service Overview**.

1.2 API Calling

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

Additionally, CS offers Java software development kits (SDKs). For details about how to use SDKs, see **Cloud Stream Service SDK Reference**.

1.3 Endpoints

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

The following table lists CS endpoints. Select a desired one based on the service requirements.

Region	Endpoint Region	Endpoint
CN North-Beijing1	cn-north-1	cs.cn-north-1.myhuaweicloud.com
CN North-Beijing4	cn-north-4	cs.cn-north-4.myhuaweicloud.com
CN East-Shanghai2	cn-east-2	cs.cn-east-2.myhuaweicloud.com
CN South-Guangzhou	cn-south-1	cs.cn-south-1.myhuaweicloud.com

1.4 Constraints

- The number of CS resources that you can create is determined by your quota. For details, see **Service Quota**.
- For more constraints, see API description.

1.5 Concepts

- Account

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

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

  An IAM user can view the account ID and user ID on the **My Credentials** page of the console. 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

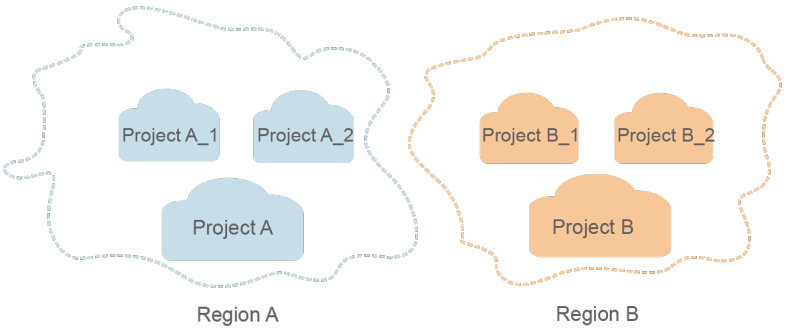
  An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an

AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to support cross-AZ high-availability systems.

- Project

  Projects group and isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each region, and subprojects can be created under each default project. IAM users can be granted permissions to access all resources in a specific project. For more refined access control, create subprojects under a project and apply for resources in the subprojects. IAM users can then be assigned permissions to access only specific resources in the subprojects.

  Figure 1-1 Project isolating model



2 API Overview

CS provides self-developed APIs. You can call the CS APIs listed in the following table to perform related functions.

Table 2-1 CS APIs

Type	API	Description
APIs related to job management	Registering to Use CS	Registers with the cloud platform to use CS.
	Granting OBS Operation Rights to CS	Grants CS the permission to access OBS buckets for saving job checkpoints and run logs.
	Collecting Summary Statistics	Collects summary statistics on users' jobs and prices.
	Querying the Current Release Version of CS	Queries the current release version of CS
	Creating a SQL Job	Uses the POST method to submit a stream SQL job in JSON format.
	Updating a SQL Job	Updates a SQL job.
	Creating a User-Defined Job	Creates user-defined jobs, which currently support the JAR format and run in exclusive clusters.
	Updating a User-Defined Job	Updates a user-defined job.
	Running a Job	Runs a specified job.
	Running Jobs in Batches	Runs specified jobs in batches.
	Querying the List of Jobs	Queries the list of jobs.

Type	API	Description
	Querying Job Details	Queries details about a specified job.
	Querying the Job Execution Plan	Queries the execution plan of a specified job.
	Querying Audit Logs	Queries the audit logs of a specified job.
	Querying Job Monitoring Information	Queries job monitoring information. You can query monitoring information about multiple jobs at the same time.
	Querying the APIG Address of a Job	Queries the APIG address of a job.
	Stopping a Job	Stops a specified job.
	Stopping Jobs in Batches	Stops specified jobs in batches.
	Deleting a Job	Deletes a specified job.
	Deleting Jobs in Batches	Deletes specified jobs in batches.
APIs related to job template management	Creating a Template	Adds a custom template.
	Updating a Template	Updates a specified template.
	Deleting a Template	Deletes a specified template.
	Querying the Template List	Queries the list of job templates. Currently, only custom templates can be queried.
APIs related to cluster management	Creating an Exclusive Cluster	Creates an exclusive cluster.
	Querying Information About Exclusive Clusters	Queries information about exclusive clusters.
	Deleting an Exclusive Cluster	Deletes an exclusive cluster.
	Querying the Cluster List of a Tenant	Queries the cluster list of a tenant.
	Updating an Exclusive Cluster	Updates an exclusive cluster.
	Terminating an Exclusive Cluster	Terminates an exclusive cluster.

Type	API	Description
	Restarting an Exclusive Cluster	Restarts an exclusive cluster.
	Querying the List of Jobs Running in an Exclusive Cluster	Queries the list of jobs running in an exclusive cluster.
	Add an IP-Domain Mapping	Adds an IP-domain mapping for a specified cluster.
	Updating the IP-Domain Mapping	Updates a specified IP-domain mapping.
	Querying an IP-Domain Mapping	Queries information about the IP-domain mapping in a specified cluster.
	Deleting an IP-Domain Mapping	Deletes the IP-domain mapping.
	Adding the hosts File	Adds an IP-domain mapping for a specified cluster through file uploading. The **hosts_file** parameter is involved in this API, and the parameter must be in the "form-data" format.
	Obtaining User Quota Information About a Tenant	Obtains user quota information about a tenant.
	Querying Quota Information About a Specified User	Queries quota information about a specified user.
	Updating Quota Information About a Specified User	Updates quota information about a specified user.
	Creating a VPC Peering Connection	Creates a VPC peering connection used to connect to other VPCs.
	Querying the List of VPC Peering Connections	Queries the list of VPC peering connections.
	Querying a VPC Peering Connection	Queries a specified VPC peering connection.
	Deleting a VPC Peering Connection	Deletes a specified VPC peering connection.
	Accepting a VPC Peering Connection Request	Accepts a VPC peering connection request.

Type	API	Description
	Creating a Route to the Local VPC	Creates a route to the local end of a VPC peering connection.
	Creating a Route to the Peer VPC	Creates a route to the peer end of a VPC peering connection.
	Query the Route List	Queries the list of routes of a VPC peering connection.
	Deleting a Route	Deletes a specific route.

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.

Table 3-1 URI parameter description

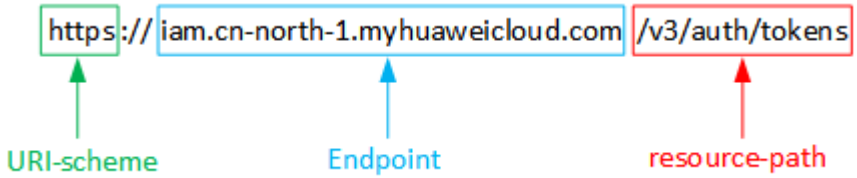
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of IAM in the **CN North-Beijing1** region is **iam.cn-north-1.myhuaweicloud.com**.
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**.

Parameter	Description
query-string	Query parameter, which is optional. Ensure that a question mark (**?**) is included before a 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.

For example, to obtain an IAM token in the **CN North-Beijing1** region, obtain the endpoint of IAM (**iam.cn-north-1.myhuaweicloud.com**) for this region and the **resource-path** (**/v3/auth/tokens**) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-1 Example URI



NOTE

To simplify the URI display in this document, 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.

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.

Method	Description
PATCH	Requests the server to update partial content 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.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

Request Header

You can also add additional 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 3-3 lists common request header fields.

Table 3-3 Common request header fields

Field	Description	Mandatory	Example
Host	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 authentication using AK/SK.	code.test.com or code.test.com:443

Field	Description	Mandatory	Example
Content-Type	Request body MIME type. You are advised to use the default value **application/json**. For APIs used to upload objects or images, the value can vary depending on the flow type.	Yes	application/json
Content-Length	Length of the request body. The unit is byte.	No	3495
X-Project-Id	Project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	User token. The user token is a response to the API used to **obtain a user token**. This API is the only one that does not require authentication. The token is the value of **X-Subject-Token** in the response.	No This field is mandatory for authentication using tokens.	The following is part of an example token: MIIPAgYJKoZIhvcNAQc-Co...ggg1BBIINPXsidG9rZ

NOTE

In addition to supporting token-based authentication, public cloud APIs also support authentication using AK/SK. During AK/SK-based authentication, an SDK is used to sign the 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 more information, 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.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

(Optional) Request Body

The request body is optional. The body of a request is often sent in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The request body transfers content except the request header. If the request body contains Chinese characters, these characters must be coded in UTF-8.

The 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*** (account name, ********* (login password), and ***xxxxxxxxxxxxxxxxxx*** (project ID) with the actual values. Obtain a project ID from **Regions and Endpoints**the administrator.

NOTE

The **scope** parameter specifies where a token takes effect. In the following example, the token takes effect only for the resources in a specified project. You can set **scope** to an account or a project under an account. For details, see **Obtaining a User Token**.

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

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

Requests for calling an API can be authenticated using either of the following methods:

- 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. Authentication using a token adds the token in a request as its header during API calling to obtain permissions to operate APIs through IAM.

In **Making an API Request**, the process of calling the API for **obtaining a user token** is described. 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:

```
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 to sign requests based on the signature algorithm or use the signing SDK to sign requests. For details about how to sign requests and use the signing SDK, see **API Request Signing Guide**.

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 more information, see **Status Code**.

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 3-2 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 3-2 Header fields of the response to the request for obtaining a user token

Response Body

The body of a response is often returned in structured format (for example, JSON or XML) 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": "cn-north-1",
.....
```

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

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

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

4 Getting Started

This section describes how to use CS APIs. Calling APIs to manage a cluster is used as example. The cluster management procedure is as follows:

1. Obtain the user token, which will be put into the request header for authentication in a subsequent request. For details about how to obtain the token, see **Authentication**.
2. Call the API for creating a SQL job to submit a streaming SQL job in POST mode. For details, see **Creating a SQL Job**.
3. Call the API for running jobs. For details, see **Running a Job**.
4. Call the API for querying the job list to obtain job information. For details, see **Querying the List of Jobs**.
5. Call the API for stopping jobs. For details, see **Stopping a Job**.

Prerequisites

- You have obtained the endpoints of IAM and CS. For details, see **Regions and Endpoints**.
- You have obtained the project ID. For details, see **Obtaining a Project ID**.

Cluster Management

The following values are examples (replace them based on the actual situation).

- IAM endpoint: **iam_endpoint**
- CS endpoint: **cs_endpoint**
- Project ID: **9bc552e6-19af-4326-800d-281a92984636**

Perform the following operations to manage the clusters:

Step 1 Before calling other APIs, obtain the token and set it as an environment variable.

```
curl -H "Content-Type:application/json" https://{iam_endpoint}/v3/auth/tokens -X POST -d '{
   "auth": {
      "identity": {
         "methods": [
            "password"
         ],
         "password": {
            "user": {
               "name": "testname",
```



```
                "domain": {
                    "name": "testname"
                },
                "password": "Passw0rd"
            }
        }
    },
    "scope": {
        "project": {
            "name": "cn-north-1"



        }
    }
  }
}' -v -k
```

1. Obtain the value of **X-Subject-Token** (indicating the **Token**) from the response header. The following is an example:
   X-Subject-Token:MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...
2. Run the following command to set the token as an environment variable:

   export Token={X-Subject-Token}

   X-Subject-Token is the token obtained in the preceding step. The following is an example command:

   ```
   export X-Auth-Token=MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...
   ```

Step 2 Call the API for creating a SQL job.

```
curl -X POST -H 'Content-Type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '
{
    "name": "my job",
    "desc": "This is a job used for counting characters.",
    "sql_body": "select * from source_table",
    "run_mode": "shared_cluster",
    "spu_number": 1,
    "parallel_number": 1,
    "checkpoint_enabled": false,
    "checkpoint_mode": "exactly_once",
    "checkpoint_interval": 0,
    "log_enabled": false
}' https://{cs_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/sql_job -v -k
```

The response is as follows:

```
STATUS CODE 200
{
    "message_id": "CS.12000",
    "message": "A CS job is created successfully.",
    "payload": {
        "job_id": 50320,
        "status_name": "job_init",
        "status_desc": ""
    },
    "current_time": 1533686888000
}
```

Step 3 Call the API for running jobs.

```
curl -X POST -H 'Content-Type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '[50320]
' https://{cs_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/job/run -v -k
```

The response is as follows:

```
STATUS CODE 200
{
    "message_id": "CS.12000",
    "message": "A CS job is created successfully.",
```



```
    "payload": {
        "job_id": 50320,
        "status_name": "job_init",
        "status_desc": ""
    },
    "current_time": 1533686888000
}
```

Step 4 Call the API for querying the job list to obtain job information.

```
curl -X GET -H 'Content-Type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://
{cs_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/jobs -k –v
```

The response is as follows:

```
{
    "message_id": "CS.23000",
    "message": "Querying of the job list succeeds.",
    "payload": {
        "total": 1,
        "jobs": [
            {
                "job_id": 50320,
                "user_id": "ac4eaa303639409c8ab099d55eb1538e",
                "cluster_id": 100000,
                "project_id": "5a3314075bfa49b9ae360f4ecd333695",
                "name": "my job",
                "desc": "This is a job used for counting characters.",
                "sql_body": "",
                "run_mode": "shared_cluster",
                "job_type": "flink_jar_job",
                "spu_number": 1,
                "parallel_number": 1,
                "job_config": {
                    "checkpoint_enabled": false,
                    "checkpoint_interval": 10,
                    "checkpoint_mode": "exactly_once",
                    "log_enabled": false
                },
                "status": "job_running",
                "status_desc": "",
                "create_time": 1516952710040,
                "update_time": 1516952770835
            }
        ]
    },
    "current_time": 1533686888000
}
```

If **status** is **job_submitting**, the job is being submitted. If **status** is **job_running**, the job is running.

Step 5 Call the API for stopping jobs.

```
curl -X POST -H "Content-Type:application/json" -H "X-Auth-Token:$Token"  -d '[50320]'
 https://{cs_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/job/stop -k -v
```

The response is as follows:

```
[ {   "message_id": "CS.16000",   "message": "The CS job stops successfully.",   "current_time":
1533686888000 } ]
```

----End

5 APIs

5.1 APIs Related to Job Management

5.1.1 Registering to Use CS

Function

This API is used to register with the cloud platform to use CS.

URI

- URI format
  POST /v1.0/{project_id}/user_register
- Parameter description

  Table 5-1 URI parameter description

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

Request

None

Response

- Example response

```
{
  "message_id": "CS.10001",
"message": "Message content",
```

```
  "current_time": 1533686888000
}
```

- Parameter description

Table 5-2 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-3 describes the status code.

Table 5-3 Status code

Status Code	Description
200	Registration succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.2 Granting OBS Operation Rights to CS

Function

This API allows users to proactively grant operation rights on OBS buckets to CS to save checkpoint setting and run logs of users' jobs.

URI

- URI format

  POST /v1.0/{project_id}/obs_authorize

- Parameter description

Table 5-4 URI parameter description

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

Request

Grant read and write permissions on OBS buckets to CS. The settings of related request parameters are in JSON format.

- Example request
```
["bucket1"]
```
- Parameter description

Table 5-5 Request parameter

Parameter	Mandatory	Type	Description
buckets	No	String	List of OBS buckets.

Response

- Example response
```
{
"message_id": "CS.10001",
"message": "Message content",
  "current_time": 1533686888000
}
```
- Parameter description

Table 5-6 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-7 describes the status code.

Table 5-7 Status code

Status Code	Description
200	Authorization succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.3 Collecting Summary Statistics

Function

This API is used to collect summary statistics on users' jobs and prices.

URI

- URI format

  GET /v1.0/{project_id}/overview
- Parameter description

  Table 5-8 URI parameter description

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

Request

None

Response

- Example response

```
{
"message_id": "CS.10001",
"message": "Message content",
  "payload": {
  "total_cost": 100,
    "job_overview": {
    "total_jobs": 100,
    "running_jobs_count": 100,
    "finished_job_count": 100,
    "exception_job_count": 100,
    "running_job_charge": 100,
    "running_job_price": 100,
    "running_job_total_spu": 100,
    "running_job_total_time": 1.5,
    "billing_unit": "CNY",
    "time_unit": "HOUR"
```

```
  },
"cluster_overview": {
      "total_clusters": 100,
      "running_cluster_count": 100,
      "exception_cluster_count": 100,
      "other_cluster_count": 100,
      "running_cluster_charge": 100,
      "running_cluster_price": 100,
      "running_cluster_total_spu": 100,
      "running_cluster_total_time": 100,
      "billing_unit": "CNY",
      "time_unit": "HOUR"
    }
  },
  "current_time": 1533686888000
}
```

- Parameter description

Table 5-9 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
payload	No	None	Information about summary statistics.
total_cost	No	Double	Total price.
job_overview	No	None	Entity for collecting summary statistics of jobs.
total_jobs	No	Int	Total number of jobs.
running_jobs_count	No	Int	Number of running jobs.
finished_job_count	No	Int	Number of completed jobs.
exception_job_count	No	Int	Number of jobs experiencing exceptions.
running_job_charge	No	Double	Total price charged for running jobs.
running_job_price	No	Double	Sum of prices per hour charged for all running jobs.
running_job_total_spu	No	Int	Total number of SPUs consumed by running jobs.
running_job_total_time	No	Double	Total running duration of running jobs.
billing_unit	No	String	Settlement currency.
time_unit	No	String	Time unit.

Parameter	Mandatory	Type	Description
cluster_overview	No	None	Entity for collecting summary statistics of clusters.
running_cluster_price	No	Double	Sum of prices per hour charged for all running clusters.
running_cluster_count	No	Int	Total number of running clusters.
running_cluster_total_time	No	Double	Total running duration of running clusters.
total_clusters	No	Int	Total number of clusters
running_cluster_total_spu	No	Int	Total number of SPUs consumed by running clusters.
running_cluster_charge	No	Double	Total price charged for running clusters.
exception_cluster_count	No	Int	Number of completed clusters.
other_cluster_count	No	Int	Number of abnormal clusters.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-10 describes the status code.

Table 5-10 Status code

Status Code	Description
200	Querying summary statistics succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.4 Querying the Current Release Version of CS

Function

This API is used to query the current release version of CS

URI

- URI format

  GET /versions

- Parameter description

  Table 5-11 URI parameter description

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

Request

None

Response

- Example response

```
{
 "id": "v1.0",
 "version": "v1.0.1",
 "min_version": "v1.0",
 "engines": [
  {
   "name": "flink",
   "version": "1.4.0"
  }
 ],
 "status": "CURRENT",
 "updated": "20180916013934",
 "current_millis": 4000000
}
```

- Parameter description

  Table 5-12 Response parameters

Parameter	Mandatory	Type	Description
id	No	String	CS version ID.
version	No	String	Current version of CS.
min_version	No	String	Initial CS version.
engines	No	array	Flink engine content.
engines: name	No	String	Flink engine name.
engines: version	No	String	Current version of the Flink engine.

Parameter	Mandatory	Type	Description
status	No	String	Status of the current CS version.
updated	No	String	Time when the CS version is updated.
current_millis	No	Integer	Current time, expressed by milliseconds.

Status Code

Table 5-13 describes the status code.

Table 5-13 Status Code

Status Code	Description
200	The query of the current CS release version succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.5 Creating a SQL Job

Function

This API is used to use the POST method to submit a stream SQL job in JSON format.

URI

- URI format

  POST /v1.0/{project_id}/sql_job
- Parameter description

  Table 5-14 URI parameter description

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

Request

Use the POST method to submit a stream SQL job in JSON format.

- Example request

```
{
"name": "myjob",
"desc": "This is a job used for counting characters.",
  "template_id": 100000,
  "cluster_id": 100000,
  "sql_body": "select * from source_table",
  "run_mode": "shared_cluster",
  "spu_number": 2,
  "parallel_number": 1,
  "checkpoint_enabled": false,
  "checkpoint_mode": "exactly_once",
  "checkpoint_interval": 0,
  "obs_bucket": "my_obs_bucket",
  "log_enabled": false,
  "smn_topic": "cs_job_exception",
  "restart_when_exception": false,
  "idle_state_retention": 0,
  "job_type": "flink_sql_job",
  "edge_group_ids": "62de1e1c-066e-48a8-a79d-f461a31b2ee1,2eb00f85-99f2-4144-bcb7-
d39ff47f9002"
}
```

- Parameter description

Table 5-15 Request parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Job name.
desc	No	String	Job description.
template_id	No	Int	Template ID. If both **template_id** and **sql_body** are specified, **sql_body** prevails. If **template_id** is specified but **sql_body** is not, fill **sql_body** with the **template_id** value.
cluster_id	No	Int	Resource ID of an exclusive cluster. Ensure that the current user has been authorized to use resources of the exclusive cluster.
sql_body	No	String	Stream SQL statement, which includes at least the following three parts: source, query, and sink.

Parameter	Mandatory	Type	Description
run_mode	Yes	String	Job running mode. The options are as follows: • **shared_cluster**: indicates that the job is running on a shared cluster. • **exclusive_cluster**: indicates that the job is running on an exclusive cluster. • **edge_node**: indicates that the job is running on an edge node.
spu_number	No	Int	Number of SPUs selected for a job.
parallel_number	No	Int	Number of parallel jobs set by a user.
checkpoint_enabled	No	Boolean	Whether to enable the automatic job snapshot function. • **true**: indicates to enable the automatic job snapshot function. • **false**: indicates to disable the automatic job snapshot function. • Default value: **false**
checkpoint_mode	No	Int	Snapshot mode. The values include: • ExactlyOnce • AtLeastOnce
checkpoint_interval	No	Int	Snapshot interval, expressed by seconds.
obs_bucket	No	String	OBS path where users are authorized to save the snapshot. This parameter is valid only when **checkpoint_enabled** is set to **true**.
log_enabled	No	Boolean	Whether to enable the function of uploading job logs to users' OBS buckets.
smn_topic	No	String	If a job fails to work, CS pushes alarm information to the SMN topic.

Parameter	Mandatory	Type	Description
restart_when_exception	No	Boolean	Whether to enable the function of automatically restarting a job upon job exceptions.
idle_state_retention	No	Int	How long the state of a key is retained without being updated before it is removed in GroupBy or Window.
job_type	No	String	Job type. This parameter can be set to **flink_sql_job** or **flink_sql_edge_job**. • If **run_mode** is set to **edge_node**, this parameter must be **flink_sql_edge_job**. • If **run_mode** is set to **shared_cluster** or **exclusive_cluster**, this parameter must be **flink_sql_job**.
edge_group_ids	No	String	List of edge computing group IDs. Use commas (,) to separate multiple IDs.
tags	No	String	Job tag.
sys_enterprise_project_id	No	String	ID of the enterprise project to which a job belongs.
udf_jar_url	No	String	OBS address where the SQL job UDF JAR file is stored.
dirty_data_strategy	No	String	Dirty data policy of a job. • **2:obs-wan-wulan3/jobs**: Save. • **1**: Trigger a job exception. • **0**: Ignore.

Response

- Example response

```
{
"message_id": "CS.10001",
"message": "Message content",
  "current_time": 1533686888000,
  "payload": {
    "job_id": 50320,
    "status_name": "running",
    "status_desc": "Status description"
  }
}
```

- Parameter description

Table 5-16 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the job status.
job_id	No	Long	Job ID.
status_name	No	String	Name of job status.
status_desc	No	String	Description of current job status, including the cause of abnormal status and suggestions.

Status Code

Table 5-17 describes the status code.

Table 5-17 Status code

Status Code	Description
200	The Stream SQL job is submitted successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.6 Updating a SQL Job

Function

The Stream SQL syntax is developed based on Apache Flink SQL. For details, see the **Cloud Stream Service SQL Syntax Reference**.

URI

- URI format

  PATCH /v1.0/{project_id}/sql_job

- Parameter description

Table 5-18 URI parameter description

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

Request

- Example request

```
{
"job_id": 50320,
  "name": "myjob",
  "desc": "My first job",
  "cluster_id": 1000000,
  "sql_body": "select * from source_table",
  "run_mode": "shared_cluster",
  "spu_number": 4,
  "parallel_number": 4,
  "checkpoint_enabled": false,
  "checkpoint_mode": "exactly_once",
  "checkpoint_interval": 10,
  "obs_bucket": "",
  "log_enabled": false,
  "smn_topic": "",
  "restart_when_exception": false,
  "idle_state_retention": 0,
  "edge_group_ids": ""
}
```

- Parameter description

Table 5-19 Request parameters

Parameter	Mandatory	Type	Description
job_id	Yes	Long	Job ID.
name	No	String	Job name.
desc	No	String	Job description.
cluster_id	No	Int	Resource ID of an exclusive cluster. Ensure that the current user has been authorized to use resources of the exclusive cluster.
sql_body	No	String	Stream SQL statement, which includes at least the following three parts: source, query, and sink.

Parameter	Mandatory	Type	Description
run_mode	No	Int	Job running mode. The options are as follows: • **shared_cluster**: indicates that the job is running on a shared cluster. • **exclusive_cluster**: indicates that the job is running on an exclusive cluster. • **edge_node**: indicates that the job is running on an edge node.
spu_number	No	Int	Number of SPUs selected for a job.
parallel_number	No	Int	Number of parallel jobs set by a user.
checkpoint_enabled	No	Boolean	Whether to enable the automatic job snapshot function. • **true**: indicates to enable the automatic job snapshot function. • **false**: indicates to disable the automatic job snapshot function. • Default value: **false**
checkpoint_mode	No	Int	Snapshot mode. The values include: Value **1** indicates that the event is processed only once. Value **2** indicates that the event is processed at least twice.
checkpoint_interval	No	Int	Snapshot interval, expressed by seconds.
obs_bucket	No	String	OBS path where users are authorized to save the snapshot. This parameter is valid only when **is_checkpoint** is set to **true**.
log_enabled	No	Boolean	Whether to enable the function of uploading job logs to users' OBS buckets.

Parameter	Mandatory	Type	Description
smn_topic	No	String	If a job fails to work, CS pushes alarm information to the SMN topic.
restart_when_exception	No	Boolean	Whether to enable the function of automatically restarting a job upon job exceptions.
idle_state_retention	No	Int	How long the state of a key is retained without being updated before it is removed in GroupBy or Window.
edge_group_ids	No	String	List of edge computing group IDs. Use commas (,) to separate multiple IDs.
tags	No	String	Job tag.
sys_enterprise_project_id	No	String	ID of the enterprise project to which a job belongs.
udf_jar_url	No	String	OBS address where the SQL job UDF JAR file is stored.
dirty_data_strategy	No	String	Dirty data policy of a job. • **2:obs-wan-wulan3/jobs**: Save. • **1**: Trigger a job exception. • **0**: Ignore.

Response

- Example response

```
{
"message_id": "CS.10001",
  "message": "The job is updated successfully.",
    "payload": {
    "update_time": 1516952770835
  },
"current_time": 1533686888000
}
```

- Parameter description

Table 5-20 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.

Parameter	Mandatory	Type	Description
message	No	String	Message content.
update_time	No	Int	Message update time, expressed by milliseconds.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about job update.
update_time	No	Int	Job update time, expressed by milliseconds.

Status Code

Table 5-21 describes the status code.

Table 5-21 Status code

Status Code	Description
200	The Stream SQL job is updated successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.7 Creating a User-Defined Job

Function

This API is used to create user-defined jobs, which currently support the JAR format and run in exclusive clusters. Table 5-22 lists the supported parameters. Parameters must be in the "form-data" format.

Table 5-22 Parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Job name.
desc	Yes	String	Job description.

Parameter	Mandatory	Type	Description
job_type	Yes	String	Job type. The options are as follows: ● **flink_jar_job**: indicates a user-defined Flink job. ● **spark_streaming_jar_job**: indicates a user-defined Spark Streaming job.
cluster_id	Yes	Integer	Resource ID of an exclusive cluster. Ensure that the current user has the permission of using resources of the exclusive cluster.
spu_number	No	Integer	Number of SPUs selected by a user for a job. You need to set this parameter when submitting a user-defined Flink job, but not when submitting a user-defined Spark job.
manager_spu	Yes	Integer	Number of management node SPUs selected by a user for a job. For user-defined Flink jobs, the parameter value is the number of Flink job managers. For user-defined Spark jobs, the parameter value is the number of Spark drivers.
parallel_number	No	Integer	Number of parallel tasks selected by a user for a job. You need to set this parameter when submitting a user-defined Flink job, but not when submitting a user-defined Spark job.
executor_number	No	Integer	Number of executors selected by a user for a Spark job. You need to set this parameter when submitting a user-defined Spark job, but not when submitting a user-defined Flink job.
executor_spu	No	Integer	Number of SPUs used by an executor in a job. You need to set this parameter when submitting a user-defined Spark job, but not when submitting a user-defined Flink job.
log_enabled	No	Boolean	Whether to enable the job log function. ● **true**: indicates to enable the job log function. ● **false**: indicates to disable the job log function. ● Default value: **false**
obs_bucket	No	String	OBS path where users are authorized to save logs when **log_enabled** is set to **true**.
smn_topic	No	String	If a job fails to work, CS pushes alarm information to the SMN topic.
jar	No	File	JAR file uploaded by the user. This parameter has a higher priority than the **jar_url** parameter.

Parameter	Mandatory	Type	Description
jar_url	No	String	OBS path of the JAR package uploaded by the user.
config	No	File	Configuration file uploaded by the user. This parameter has a higher priority than the **config_url** parameter.
config_url	No	String	OBS path of the configuration file uploaded by the user.
main_class	No	String	Job entry class.
args	No	String	Job entry parameter.
restart_when_exception	No	Boolean	Whether to enable restart upon exception. The default value is **false**.
tags	No	String	Job tag.
sys_enterprise_project_id	No	String	ID of the enterprise project to which a job belongs.

URI

- URI format

  POST /v1.0/{project_id}/jar_job
- Parameter description

  Table 5-23 URI parameter description

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

Request

The request format is form-data. For details about the parameter configuration, see **Table 5-22**.

Response

- Example response

```
{
"message_id": "CS.10001",
 "message": "A job is created successfully.",
```



```
  "current_time": 1533686888000,
  "payload": {
    "job_id": 50320,
    "status_name": "running",
    "status_desc": "Status description"
  }
}
```

- Parameter description

Table 5-24 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the job status.
job_id	Yes	Long	Job ID.
status_name	No	String	Name of job status.
status_desc	No	String	Status description.

Status Code

Table 5-25 describes the status code.

Table 5-25 Status code

Status Code	Description
200	A JAR job is created successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.8 Updating a User-Defined Job

Function

This API is used to update user-defined jobs, which currently support the JAR format and run in exclusive clusters. Table 5-26 lists the supported parameters. Parameters must be in the "form-data" format.

Table 5-26 Parameter description

Parameter	Mandatory	Type	Description
job_id	Yes	Integer	Job ID.
name	No	String	Job name.
desc	No	String	Job description.
cluster_id	No	Integer	Resource ID of an exclusive cluster. Ensure that the current user has the permission of using resources of the exclusive cluster.
spu_number	No	Integer	Number of SPUs selected by a user for a job. You need to set this parameter when submitting a user-defined Flink job, but not when submitting a user-defined Spark job.
manager_spu	Yes	Integer	Number of management node SPUs selected by a user for a job. For user-defined Flink jobs, the parameter value is the number of Flink job managers. For user-defined Spark jobs, the parameter value is the number of Spark drivers.
parallel_number	No	Integer	Number of parallel tasks selected by a user for a job. You need to set this parameter when submitting a user-defined Flink job, but not when submitting a user-defined Spark job.
executor_number	No	Integer	Number of executors selected by a user for a Spark job. You need to set this parameter when submitting a user-defined Spark job, but not when submitting a user-defined Flink job.
executor_spu	No	Integer	Number of SPUs used by an executor in a job. You need to set this parameter when submitting a user-defined Spark job, but not when submitting a user-defined Flink job.
log_enabled	No	Boolean	Whether to enable the job log function. ● **true**: indicates to enable the job log function. ● **false**: indicates to disable the job log function. ● Default value: **false**
obs_bucket	No	String	OBS path where users are authorized to save logs when **log_enabled** is set to **true**.

Parameter	Mandatory	Type	Description
smn_topic	No	String	If a job fails to work, CS pushes alarm information to the SMN topic.
jar	No	File	JAR file uploaded by the user. This parameter has a higher priority than the **jar_url** parameter.
jar_url	No	String	OBS path of the JAR package uploaded by the user.
config	No	File	Configuration file uploaded by the user. This parameter has a higher priority than the **config_url** parameter.
config_url	No	String	OBS path of the configuration file uploaded by the user.
main_class	No	String	Job entry class.
args	No	String	Job entry parameter.
restart_when_exception	No	Boolean	Whether to enable restart upon exception. The default value is **false**.
tags	No	String	Job tag.
sys_enterprise_project_id	No	String	ID of the enterprise project to which a job belongs.

URI

- URI format

  PATCH /v1.0/{project_id}/jar_job
- Parameter description

  Table 5-27 URI parameter description

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

Request

The request format is form-data, and the job ID is mandatory. For details about the parameter configuration, see **Table 5-26**.

Response

- Example response

```
{
"message_id": "CS.10001",
 "message": "The job is updated successfully.",
 "current_time": 1533686888000,
 "payload": {
  "update_time": 1516952770835
 }
}
```

- Parameter description

Table 5-28 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the job to be created.
update_time	No	Int	Job update time, expressed by milliseconds.

Status Code

Table 5-29 describes the status code.

Table 5-29 Status code

Status Code	Description
200	The user-defined job is updated successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.9 Running a Job

Function

This API is used to run jobs.

URI

- URI format
  POST /v1.0/{project_id}/job/{job_id}/run
- Parameter description

  Table 5-30 URI parameter description

Parameter	Type	Mandatory	Description
project_id	Route parameter	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
job_id	Route parameter	Yes	Job ID.
resumeSavePoint	Query parameter	No	Whether to restore a job from the latest savepoint. The data type is Boolean. • If this parameter is set to **true**, the job is restored from the latest savepoint. • If this parameter is set to **false**, the job is started normally, not from a specific savepoint. The default value is **false**.

Request

None

Response

- Example response

```
{
"message_id": "CS.10001",
"message": "Message content",
```

```
  "current_time": 1533686888000
}
```

- Parameter description

Table 5-31 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-32 describes the status code.

Table 5-32 Status code

Status Code	Description
200	The request of starting a job is sent successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.10 Running Jobs in Batches

Function

This API is used to run jobs in batches.

URI

- URI format
  POST /v1.0/{project_id}/job/run
- Parameter description

Table 5-33 URI parameter description

Parameter	Type	Mandatory	Description
project_id	Route parameter	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
job_id	Route parameter	Yes	Job ID.
resumeSavePoint	Query parameter	No	Whether to restore a job from the latest savepoint. The data type is Boolean. • If this parameter is set to **true**, the job is restored from the latest savepoint. • If this parameter is set to **false**, the job is started normally, not from a specific savepoint. The default value is **false**.

Request

- Example request

```
[
   298765,
   298766
]
```

- Parameter description

Table 5-34 Request parameter

Parameter	Mandatory	Type	Description
job_id	Yes	Long	Job ID.

Response

- Example response

```
{
"message_id": "CS.10001",
"message": "Message content",
  "current_time": 1533686888000
}
```

- Parameter description

Table 5-35 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-36 describes the status code.

Table 5-36 Status code

Status Code	Description
200	The request of starting a job is sent successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.11 Querying the List of Jobs

Function

This API is used to query the list of the current user's jobs. You can set the job ID as the ID and query jobs whose IDs are greater than or less than the ID. You can also query jobs in specific status, for example, in running status or other. By default, all jobs are queried.

The following table lists parameters involved in the API.

Table 5-37 Parameter description

Parameter	Description
name	Job name.
status	Status code of a job. For details, see **Table 5-40**.
cluster_id	ID of an exclusive cluster.
show_detail	Whether to return the job details.
cursor	Job ID.

Parameter	Description
next	Whether to switch to the previous or next page.
limit	Number of returned data records.
order	Query result display. Value **asc** indicates that the query results are displayed in ascending order, and value **desc** indicates that the query results are displayed in descending order.
root_job_id	Edge parent job ID, which is used to query sub-jobs of a specified edge job. If this parameter is not specified, all non-edge jobs and edge parent jobs, instead of edge sub-jobs, are queried.

URI

- URI format

  GET /v1.0/{project_id}/jobs
- Parameter description

  Table 5-38 URI parameter description

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

Request

None

Response

- Example response

```
{
   "message_id": "CS.10001",
   "message": "Querying of the job list succeeds.",
   "current_time": 1533686888000,
   "payload": {
   "total": 1,
   "jobs": [
    {
      "job_id": 50320,
      "name": "my_job_001",
      "desc": "This is my job.",
      "username": "cs_testuser",
      "job_type": "flink_jar_job",
      "status": "job_running",
      "status_desc": "",
```

```
      "create_time": 1516952710040,
      "start_time": 1516952710740,
      "duration": 6838266,
      "user_id": "ac4eaa303639409c8ab099d55eb1538e",
      "cluster_id": 100000,
      "project_id": "5a3314075bfa49b9ae360f4ecd333695",
      "sql_body": "select * from source_table",
      "run_mode": "shared_cluster",
      "spu_number": 0,
      "parallel_number": 0,
      "job_config": {
        "checkpoint_enabled": true,
        "checkpoint_mode": "exactly_once",
        "checkpoint_interval": 0,
        "log_enabled": true,
        "obs_bucket": "obs-demo",
        "smn_topic": "cs_job_exception",
        "root_id": 0,
        "edge_group_ids": "62de1e1c-066e-48a8-a79d-f461a31b2ee1,2eb00f85-99f2-4144-bcb7-
d39ff47f9002"
      },
      "jar_url": "cs-flink-jar/52818/WindowJoin.jar",
      "main_class": "org.apache.spark.examples.streaming.JavaQueueStream",
      "args": "",
      "execution_graph": "",
      "update_time": 1516952770835
    }
  ]
 }
 }
```


- Parameter description

Table 5-39 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about a job list.
total	No	Int	Number of records in the query result.
jobs	No	None	Information about a job.
job_id	Yes	Long	Job ID.
name	No	String	Job name.
desc	No	String	Job description.
username	No	String	Username. This parameter is valid only when **show_detail** is set to **true**.
job_type	No	String	Job type.

Parameter	Mandatory	Type	Description
status	No	String	Job status.
status_desc	No	String	Description of job status.
create_time	Yes	Int	Time when a job is created.
start_time	No	Int	Time when a job is enabled. Value **0** indicates that the job has not been enabled.
duration	No	Int	Running duration of a job, expressed by milliseconds.
user_id	No	String	ID of the user who creates the job. This parameter is valid only when **show_detail** is set to **true**.
cluster_id	No	Int	Resource ID of an exclusive cluster. Ensure that the current user has been authorized to use resources of the exclusive cluster.
project_id	No	String	ID of the project to which a job belongs. This parameter is valid only when **show_detail** is set to **true**.
sql_body	No	String	Stream SQL statement.
run_mode	No	String	Job running mode when **show_detail** is set to **true**: • **shared_cluster**: indicates that the job is running on a shared cluster. • **exclusive_cluster**: indicates that the job is running on an exclusive cluster. • **edge_node**: indicates that the job is running on an edge node.
spu_number	No	Int	Number of SPUs selected for a job.

Parameter	Mandatory	Type	Description
parallel_number	No	Int	Number of parallel jobs set by a user.
job_config	No	None	Job configuration. This parameter is valid only when **show_detail** is set to **true**.
checkpoint_enabled	No	boolean	Whether to enable the automatic job snapshot function. • **true**: indicates to enable the automatic job snapshot function. • **false**: indicates to disable the automatic job snapshot function. • Default value: **false**
checkpoint_mode	No	String	Snapshot mode. The values include: • ExactlyOnce • AtLeastOnce
checkpoint_interval	No	Int	Snapshot interval, expressed by seconds.
log_enabled	No	boolean	Whether to enable the log saving function.
obs_bucket	No	String	Name of an OBS bucket.
smn_topic	No	String	If a job fails to work, CS pushes alarm information to the SMN topic.
root_id	No	Int	Parent job ID. This parameter is valid only when **show_detail** is set to **false**.
edge_group_ids	No	String	List of edge computing group IDs. Use commas (,) to separate multiple IDs. This parameter is valid only when **show_detail** is set to **false**.
checkpoint_interval	No	Int	Snapshot interval, expressed by seconds.

Parameter	Mandatory	Type	Description
jar_url	No	String	OBS path of the JAR package. This parameter is valid only when **show_detail** is set to **true**.
main_class	No	String	OBS path where users are authorized to save the snapshot when **checkpoint_enabled** is set to **true**. This parameter is valid only when **show_detail** is set to **true**.
args	No	String	Job running parameter of the JAR package. This parameter is valid only when **show_detail** is set to **true**.
execution_graph	No	String	Job execution plan. This parameter is valid only when **show_detail** is set to **true**.
update_time	No	Int	Job update time. This parameter is valid only when **show_detail** is set to **true**.

Status Code

Table 5-40 Status code

Status Code	Description
200	Job list query succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.12 Querying Job Details

Function

This API is used to query details of a job.

URI

- URI format

  GET /v1.0/{project_id}/job/{job_id}

- Parameter description

  Table 5-41 URI parameter description

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

Request

None

Response

- Example response

```
{
"message_id": "CS.10001",
"message": "The query of job details succeeds.",
"current_time": 1533686888000,
"payload": {
   "job_id": 50320,
   "name": "my_job_001",
   "desc": "This is my job.",
   "job_type": "flink_jar_job",
   "status": "job_running",
   "status_desc": "",
   "create_time": 1516952710040,
   "start_time": 1516952710740,
   "user_id": "ac4eaa303639409c8ab099d55eb1538e",
   "cluster_id": 100000,
   "project_id": "5a3314075bfa49b9ae360f4ecd333695",
   "sql_body": "select * from source_table",
   "run_mode": "shared_cluster",
   "spu_number": 0,
   "parallel_number": 0,
   "job_config": {
     "checkpoint_enabled": true,
     "checkpoint_mode": "exactly_once",
     "checkpoint_interval": 0,
     "log_enabled": true,
     "obs_bucket": "string",
     "smn_topic": "cs_job_exception",
     "root_id": 0,
     "edge_group_ids": "62de1e1c-066e-48a8-a79d-f461a31b2ee1,2eb00f85-99f2-4144-bcb7-
d39ff47f9002"
   },
   "jar_url": "cs-flink-jar/52818/WindowJoin.jar",
   "main_class": "org.apache.spark.examples.streaming.JavaQueueStream",
   "args": "",
   "execution_graph": "",
   "update_time": 1516952770835
 }
}
```

- Parameter description

Table 5-42 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about a job list.
job_id	No	Long	Job ID.
name	No	String	Job name.
desc	No	String	Job description.
job_type	No	String	Job type.
status	No	String	Available job statuses are as follows: • Draft • Running • Idle • Finished • Stopped • Submitting • Submission failed • Running exception • Stopping • Stopping failed • Stopped due to arrears • Restoring (recharged jobs) • Creating savepoint
status_desc	No	String	Description of job status.
create_time	No	Long	Timestamp when a job is created.
start_time	No	Long	Timestamp when a job is started.
user_id	No	Long	ID of the user who creates the job.

Parameter	Mandatory	Type	Description
cluster_id	No	Int	Resource ID of an exclusive cluster. Ensure that the current user has been authorized to use resources of the exclusive cluster.
project_id	No	String	ID of the project to which a job belongs.
sql_body	No	String	SQL content in a Stream SQL job.
run_mode	No	String	Job running mode. The options are as follows: • **shared_cluster**: indicates that the job is running on a shared cluster. • **exclusive_cluster**: indicates that the job is running on an exclusive cluster. • **edge_node**: indicates that the job is running on an edge node.
spu_number	No	Int	Number of SPUs selected for a job.
parallel_number	No	Int	Number of parallel jobs set by a user.
job_config	No	None	Job configuration. This parameter is valid only when **show_detail** is set to **true**.
checkpoint_enabled	No	Boolean	Whether to enable the automatic job snapshot function. • **true**: indicates to enable the automatic job snapshot function. • **false**: indicates to disable the automatic job snapshot function. • Default value: **false**

Parameter	Mandatory	Type	Description
checkpoint_mode	No	String	Snapshot mode. Available options are as follows: • ExactlyOnce • AtLeastOnce
checkpoint_interval	No	Int	Snapshot interval, expressed by seconds.
log_enabled	No	boolean	Whether to enable the log saving function.
obs_bucket	No	String	OBS path where users are authorized to save the snapshot. This parameter is valid only when **is_checkpoint** is set to **true**.
smn_topic	No	String	If a job fails to work, CS pushes alarm information to the SMN topic.
root_id	No	Int	Parent job ID.
edge_group_ids	No	String	List of edge computing group IDs. Use commas (,) to separate multiple IDs.
jar_url	No	String	OBS path of the JAR package.
main_class	No	String	OBS path where users are authorized to save the snapshot. This parameter is valid only when **checkpoint_enabled** is set to **true**.
args	No	String	Job running parameter of the JAR package.
execution_graph	No	String	Job execution plan.
update_time	No	Int	Time when a job is updated.

Status Code

Table 5-43 Status code

Status Code	Description
200	Querying details of a job succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.13 Querying the Job Execution Plan

Function

This API is used to query the job execution plan.

URI

- URI format

  GET /v1.0/{project_id}/job/{job_id}/execute_graph
- Parameter description

  Table 5-44 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
job_id	Yes	Job ID.

Request

None

Response

- Example response

```
{
"message_id": "CS.10001",
  "message": "Querying the job execution plan succeeds.",
  "current_time": 1533686888000,
"payload": {
    "execute_plan_items": "[{\"key\":\"value\"}]"
  }
}
```

- Parameter description

Table 5-45 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	Sting	Message type ID.
message	No	Sting	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the job execution plan.
execute_plan_items	No	Sting	Job execution plan.

Status Code

Table 5-46 Status code

Status Code	Description
200	Querying the job execution plan succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.14 Querying Audit Logs

Function

This API is used to query the list of users' audit logs. The following table lists parameters involved in the API.

Table 5-47 Parameter description

Parameter	Description
resource_type	Resource type. The parameter value can be **job**, **template**, and **cluster**.
cursor	Job ID, template ID, or cluster ID.

Parameter	Description
resource_id	Resource ID.
page_number	Page number that is queried.
limit	Maximum number of records that can be queried. A maximum of 50 records can be queried.

URI

- URI format

  GET /v1.0/{project_id}/audit_logs
- Parameter description

  Table 5-48 URI parameter description

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

Request

None

Response

- Example response

```
[
 {
  "message_id": "CS.10001",
  "message": "Querying audit logs succeeds.",
  "current_time": 1533870937432,
  "payload": {
   "total": 0,
   "traces": [
    {
     "resource_id": "10000",
     "resource_name": "my job",
     "event_name": "startJob",
     "event_type": "ConsoleAction",
     "event_rating": "normal",
     "op_user": "cs_testuser",
     "op_time": 1533869273712,
     "op_ip": "10.218.216.118",
     "op_result": "CS.14000::RUN_JOB_SUCCESS:: The request for submitting CS jobs is delivered successfully."
    }
   ]
  }
 }
]
```

- Parameter description

Table 5-49 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	Sting	Message type ID.
message	No	Sting	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the job audit log.
total	No	Int	Total number of audit log records in the query result.
traces	No	None	Information about the audit log.
resource_id	Yes	Long	Job ID.
resource_name	Yes	String	Job name.
event_name	Yes	String	Event name.
event_type	Yes	String	Event type.
event_rating	Yes	String	Event level.
op_user	Yes	String	Name of job status.
op_time	Yes	String	Event recording time.
op_ip	Yes	String	IP address of the operator.
op_result	Yes	String	Operation result.

Status Code

Table 5-50 Status code

Status Code	Description
200	The query of audit logs succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.15 Querying Job Monitoring Information

Function

This API is used to query job monitoring information. You can query monitoring information about multiple jobs at the same time.

URI

- URI format

  POST /v1.0/{project_id}/jobs/metrics

- Parameter description

  Table 5-51 URI parameter description

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

Request

The request parameter setting is in JSON format.

- Example request

```
{
   "job_ids": [298765, 298766]
}
```

- Parameter description

  Table 5-52 Request parameter

Parameter	Mandatory	Type	Description
job_ids	Yes	Long array	List of job IDs.

Response

- Example response

```
{
  "message_id": "CS.10001",
"message": "Message content",
  "current_time": 40000000,
  "payload": {
    "jobs": [
      {
        "job_id": 0,
        "metrics": {
          "sources": [
            {
```

```
            "name": "Source: KafKa_6070_KAFKA_SOURCE",
            "records": 0,
            "corrupted_records": 0
          }
        ],
        "sinks": [
          {
            "name": "Source: KafKa_6070_KAFKA_SOURCE",
            "records": 0,
            "corrupted_records": 0
          }
        ],
        "total_read_rate": 100,
        "total_write_rate": 100
      }
    }
  ]
 }
}
```


- Parameter description

Table 5-53 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Job monitoring information.
jobs	No	None	Monitoring information about all jobs.
job_id	No	Long	Job ID.
metrics	No	None	All input and output monitoring information about a job.
sources	No	None	All source streams.
sinks	No	None	All sink streams.
name	No	String	Name of the source or sink stream.
records	No	Long	Total number of records.
corrupted_records	No	Long	Number of dirty data records.
total_read_rate	No	Double	Total read rate.
total_write_rate	No	Double	Total write rate.

Status Code

Table 5-54 Status code

Status Code	Description
200	The query of job monitoring information succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.16 Querying the APIG Address of a Job

Function

This API is used to query the APIG address of a job.

URI

- URI format
  GET /v1.0/{project_id}/job/{job_id}/apig_sinks
- Parameter description

  Table 5-55 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
job_id	Yes	Job ID.

Request

None

Response

- Example response

```
{
  "message_id": "CS.10001",
"message": "Message content",
  "current_time": 3423423423,
  "payload": {
    "sinks": [
      {
        "app_id": "ee412697e20c4ab8b567bffd71724991",
        "sink_name": "CAR_INFOS_DETAIL",
```

```
      "url": "https://fb2ce37f71f04d91ac67c4efcd75a172.apigw.cn-north-1.huaweicloud.com/jobid/
CAR_INFOS_DETAIL"


    }
   ],
   "metadata": [
    {
     "name": "CAR_INFOS_DETAIL",
     "encode": "csv",
     "delimiter": ",",
     "attrs": [
      {
       "name": "car_id",
       "type": "STRING"
      }
     ]
    }
   ]
  }
}
```

- Parameter description

Table 5-56 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the job audit log.
sinks	No	None	Information about the APIG address of the job.
app_id	No	String	Application ID output by the sink stream of the job.
sink_name	No	String	Name of the sink stream of the job.
url	No	String	APIG address of the sink stream of the job.
metadata	No	None	Metadata information output by the sink stream.
name	No	String	Name of the sink stream of the job.
encode	No	String	Format of the sink stream of the job, for example, CSV.
delimiter	No	String	Delimiter of the sink stream of the job, for example, comma (,).
attrs	No	None	Attribute information of the sink stream of the job.

Parameter	Mandatory	Type	Description
name	No	String	Attribute name of the sink stream of the job.
type	No	String	Attribute type of the sink stream of the job.

Status Code

Table 5-57 Status code

Status Code	Description
200	The query of the APIG address of the job succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.17 Stopping a Job

Function

This API is used to stop a running job.

URI

- URI format

  POST /v1.0/{project_id}/job/{job_id}/stop
- Parameter description

  Table 5-58 URI parameter description

Parameter	Type	Mandatory	Description
project_id	Route parameter	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
job_id	Route parameter	Yes	Job ID.

Parameter	Type	Mandatory	Description
triggerSavePoint	Query parameter	No	Whether to create a savepoint for a job to store the job status information before stopping it. The data type is Boolean. • If this parameter is set to **true**, a savepoint is created. • If this parameter is set to **false**, no savepoint is created. The default value is **false**.

Request

None

Response

- Example response

```
{
"message_id": "CS.10001",
"message": "Message content",
  "current_time": 1533686888000
}
```

- Parameter description

Table 5-59 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-60 describes the status code.

Table 5-60 Status code

Status Code	Description
200	The request of stopping a job is sent successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.18 Stopping Jobs in Batches

Function

This API is used to stop running jobs in batches.

URI

- URI format

  POST /v1.0/{project_id}/job/stop

- Parameter description

  Table 5-61 URI parameter description

Parameter	Type	Mandatory	Description
project_id	Route parameter	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
job_id	Route parameter	Yes	Job ID.
triggerSavePoint	Query parameter	No	Whether to create a savepoint for a job to store the job status information before stopping it. The data type is Boolean. • If this parameter is set to **true**, a savepoint is created. • If this parameter is set to **false**, no savepoint is created. The default value is **false**.

Request

- Example request

```
[
  298765,
  298766
]
```

- Parameter description

Table 5-62 Request parameter

Parameter	Mandatory	Type	Description
job_id	Yes	Long	Job ID.

Response

- Example response

```
{
"message_id": "CS.10001",
"message": "Message content",
  "current_time": 1533686888000
}
```

- Parameter description

Table 5-63 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-64 describes the status code.

Table 5-64 Status code

Status Code	Description
200	The request of stopping a job is sent successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.19 Deleting a Job

Function

This API is used to delete a job in any status.

URI

- URI format

  DELETE /v1.0/{project_id}/job/{job_id}

- Parameter description

  Table 5-65 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
job_id	Yes	Job ID.

Request

None

Response

- Example response

```
{
  "message_id": "CS.18000",
  "message": "The CS job is deleted successfully.",
  "current_time": 1550218039715
}
```

- Parameter description

  Table 5-66 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-67 describes the status code.

Table 5-67 Status code

Status Code	Description
200	A job is deleted successfully.

Status Code	Description
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.1.20 Deleting Jobs in Batches

Function

This API is used to delete jobs in any status in batches.

URI

- URI format

  DELETE /v1.0/{project_id}/job

- Parameter description

  Table 5-68 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
job_id	Yes	Job ID.

Request

- Example request

```
[
  298765,
  298766
]
```

- Parameter description

  Table 5-69 Request parameter

Parameter	Mandatory	Type	Description
job_id	Yes	Long	Job ID.

Response

- Example response

```
{
  "message_id": "CS.18000",
  "message": "The CS job is deleted successfully.",
  "current_time": 1550218039715
}
```

- Parameter description

Table 5-70 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-71 describes the status code.

Table 5-71 Status code

Status Code	Description
200	A job is deleted successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.2 APIs Related to Job Template Management

5.2.1 Creating a Template

Function

This API is used to create a user template for the CS service. A maximum of 100 user templates can be created.

URI

- URI format

POST /v1.0/{project_id}/job_template

- Parameter description

Table 5-72 URI parameter description

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

Request

- Example request

```
{
"name": "simple_stream_sql",
"desc": "Example of quick start",
"sql_body": "select * from source_table"
}
```

- Parameter description

Table 5-73 Request parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Template name.
desc	No	String	Template description.
sql_body	No	String	Stream SQL statement, which includes at least the following three parts: source, query, and sink.

Response

- Example response

```
{
  "message_id": "CS.10001",
  "message": "string",
  "current_time": 1533686888000,
  "payload": {
    "template_id": 0,
    "name": "IoT_example",
"desc": "Example of quick start",
    "create_time": 1516952710040,
    "is_success": true
  }
}
```

- Parameter description

Table 5-74 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the template to be created.
template_id	No	Int	Template ID.
name	No	String	Template name.
desc	No	String	Template description.
create_time	No	Int	Time when the template is created.
is_success	No	boolean	Whether the template is successfully created.

Status Code

Table 5-75 describes the status code.

Table 5-75 Status code

Status Code	Description
200	A template is created successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.2.2 Updating a Template

Function

This API is used to update existing templates in CS.

URI

- URI format
  PATCH /v1.0/{project_id}/job_template

- Parameter description

Table 5-76 URI parameter description

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

Request

Submit a request for modifying the SQL job template.

- Example request

```
{
  "id": "10000",
  "name": "simple_stream_sql",
  "desc": "Example of quick start",
  "sql_body": "select * from source_table"
}
```

- Parameter description

Table 5-77 Request parameters

Parameter	Mandatory	Type	Description
id	Yes	Int	Template ID.
name	No	String	Template name.
desc	No	String	Template description.
sql_body	No	String	Stream SQL statement, which includes at least the following three parts: source, query, and sink.

Response

- Example response

```
{
  "message_id": "CS.18014",
  "message": "The template is updated successfully.",
  "current_time": 1550218438483
}
```

- Parameter description

Table 5-78 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-79 describes the status code.

Table 5-79 Status code

Status Code	Description
200	A template is updated successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.2.3 Deleting a Template

Function

This API is used to delete a template. A template used by jobs can also be deleted.

URI

- URI format
  DELETE /v1.0/{project_id}/job_template/{template_id}
- Parameter description

Table 5-80 URI parameter description

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

Request

None

Response

- Example response

```
{
  "message_id": "CS.18013",
  "message": "The template is deleted successfully.",
  "payload": {
    "template_id": 11,
    "is_success": true
  },
  "current_time": 1550218615961
}
```

- Parameter description

Table 5-81 Request parameters

Parameter	Mandatory	Type	Description
message_id	No	Sting	Message type ID.
message	No	Sting	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the template to be deleted.
template_id	No	Int	Template ID.
is_success	No	boolean	Whether the template is successfully deleted.

Status Code

Table 5-82 describes the status code.

Table 5-82 Status code

Status Code	Description
200	A template is deleted successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.2.4 Querying the Template List

Function

This API is used to query the job template list. Currently, only custom templates can be queried.

The following parameters are involved in this API: **cursor**, **limit**, and **order**.

Table 5-83 Parameter description

Parameter	Description
cursor	Job ID.
limit	Maximum number of records that can be queried.
order	Query result display. Value **asc** indicates that the query results are displayed in ascending order, and value **desc** indicates that the query results are displayed in descending order.

URI

- URI format

  GET /v1.0/{project_id}/job_templates
- Parameter description

  Table 5-84 URI parameter description

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

Request

None

Response

- Example response

```
{
  "message_id": "CS.18011",
  "message": "The template list is obtained successfully.",
  "payload": {
    "total": 1,
    "templates": [
```

```
    {
      "template_id": 12,
      "name": "simple_stream_sql",
"desc": "Example of quick start",
      "create_time": 1550218731820,
      "update_time": 1550218731820,
      "sql_body": "select * from source_table"
    }
  ]
 },
 "current_time": 1550218761109
}
```

- Parameter description

Table 5-85 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	Sting	Message type ID.
message	No	Sting	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the template list.
total	No	Int	Total number of templates.
templates	No	None	Detailed information about a template.
template_id	No	Int	Template ID.
name	No	String	Template name.
desc	No	String	Template description.
create_time	No	Int	Time when the template is created.
update_time	No	Int	Update time of the template.
sql_body	No	String	Stream SQL statement, which includes at least the following three parts: source, query, and sink.

Status Code

Table 5-86 describes the status code.

Table 5-86 Status code

Status Code	Description
200	Template list query succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3 APIs Related to Cluster Management

5.3.1 Creating an Exclusive Cluster

Function

This API is used to create an exclusive cluster for CS users with the **cs_adm** role. The exclusive cluster is converted to stream processing units (SPUs) for on-demand charging.

URI

- URI format

  POST /v1.0/{project_id}/reserved_cluster
- Parameter description

  Table 5-87 URI parameter description

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

Request

Create an exclusive cluster, for which the request parameters are in JSON format.

- Example request

```
{
  "name": "mycluster",
  "desc": "This is my cluster.",
  "maximum_spu_quota": 100,
  "vpc_cidr": "172.16.0.0/23",
  "subnet_cidr": "172.16.0.0/23",
  "subnet_gateway": "172.16.0.1"
}
```

- Parameter description

Table 5-88 Request parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Cluster name.
desc	No	String	Cluster description.
maximum_spu_quota	No	Int	Maximum SPU quota of a cluster.
vpc_cidr	No	String	VPC network segment where the cluster is located.
subnet_cidr	No	String	Subnet network segment where the cluster is located.
subnet_gateway	No	String	Subnet gateway for the cluster.
tags	No	String	Job tag.
sys_enterprise_project_id	No	String	ID of the enterprise project to which a job belongs.
manager_node_spu	No	String	SPU specifications of management nodes.
is_arm_arch	No	Boolean	CPU type of a cluster.

Response

- Example response

```
{
  "message_id": "CS.21000",
  "message": "Cluster 1000000 is created successfully.",
  "payload": {
    "cluster_id": "1000000"
  },
  "current_time": 1550197806373
}
```

- Parameter description

Table 5-89 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
payload	No	None	Information about the exclusive cluster.

Parameter	Mandatory	Type	Description
cluster_id	No	String	Cluster ID.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-90 describes the status code.

Table 5-90 Status code

Status Code	Description
200	An exclusive cluster is successfully created.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.2 Querying Information About Exclusive Clusters

Function

This API is used to query information about exclusive clusters created by users.

URI

- URI format

  GET /v1.0/{project_id}/reserved_cluster/{cluster_id}
- Parameter description

Table 5-91 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	ID of an exclusive cluster.

Request

None

Response

- Example response

```
{
  "message_id": "CS.21010",
  "message": "Querying the cluster information succeeds.",
  "payload": {
    "cluster_id": 10,
    "name": "mycluster2",
    "desc": "Cluster description",
    "status_code": 5,
    "status_name": "cluster_running",
    "status_desc": "cluster_running",
    "spu_used": 0,
    "maximum_spu_quota": 100,

    "region_id": "cn-north-1",



    "error_reason": "",
    "created_at": 1550081520879,
    "vpc_cidr": "172.16.0.0/23",
    "subnet_cidr": "172.16.0.0/23",
    "subnet_gateway": "172.16.0.1",
    "manager_node_spu": "2",
    "sys_enterprise_project_id": "0"
  },
  "current_time": 1550199088218
}
```

- Parameter description

Table 5-92 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	Array of objects	Information about the exclusive cluster.

Table 5-93 payload parameter description

Parameter	Mandatory	Type	Description
cluster_id	No	String	Cluster ID.
name	No	String	Cluster name.

Parameter	Mandatory	Type	Description
desc	No	String	Cluster description.
status_code	No	Integer	Status code for a cluster. For details, see **Table 5-94**.
status_name	No	String	Status name of a cluster. Available cluster statuses are as follows: • Creating • ECS creation failed • Ready to start • Starting • Running • Stopping • Stop failed • Stopped • Restarting • Deleting • Deleted • Deletion failed • Arrears, Stopping • Arrears, Stop failed • Arrears stopped • Restoring (recharged cluster) • Thaw recovery failed
status_desc	No	String	Status description of a cluster.
region_id	No	String	Region where a cluster is currently located.
spu_used	No	String	SPU quota used by a cluster.
maximum_spu_quota	No	String	Maximum SPU quota of a cluster.
error_reason	No	String	Cause of the error that results in cluster status exceptions.
created_at	No	String	Time when a cluster is created.

Table 5-94 Cluster Status Code

Status Code	Status Description
1	A cluster is being created.
2	A cluster fails to be created.
3	A cluster is created successfully.
4	A cluster is starting.
5	A cluster is running.
6	A cluster is abnormal but still running.
7	A cluster is abnormal and has stopped.
8	A cluster is in the idle state.
9	A cluster is being stopped.
10	A cluster fails to be stopped.
11	A cluster has stopped.
12	A cluster is stopped successfully.
13	A cluster stops because the tenant account is in arrears.
14	A cluster that stops due to arrears is restored after the account is recharged.
15	A cluster is deleted because the tenant account is in arrears and not recharged within the validity period.
16	A cluster is being deleted.
17	A cluster fails to be deleted.
18	A cluster is being upgraded.
19	A cluster fails to be upgraded.
20	Capacity expansion of a cluster is in progress.
25	A cluster is being stopped because the tenant account is in arrears.
26	A cluster fails to stop because the tenant account is in arrears.
27	A cluster that stops due to arrears fails to be restored after the account is recharged.
28	The tenant manually restarts the cluster.
29	A cluster is deleted because the tenant has stopped the service.

Status Code

Table 5-95 describes the status code.

Table 5-95 Status code

Status Code	Description
200	Querying the information about exclusive clusters succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.3 Deleting an Exclusive Cluster

Function

This API is used to delete an exclusive cluster. If there are jobs running in the exclusive cluster to be deleted, jobs will stop automatically and immediately.

URI

- URI format

  DELETE /v1.0/{project_id}/reserved_cluster/{cluster_id}
- Parameter description

  Table 5-96 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	ID of an exclusive cluster.

Request

None

Response

- Example response

```
{
  "message_id": "CS.21020",
  "message": "Cluster 10 is being deleted.",
```

```
  "current_time": 1550199559915
}
```

- Parameter description

Table 5-97 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-98 describes the status code.

Table 5-98 Status code

Status Code	Description
200	An exclusive cluster is successfully deleted.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.4 Querying the Cluster List of a Tenant

Function

This API is used to query the list of clusters of a tenant. Following parameters are involved in this API: **name**, **status**, **cursor**, **next**, **limit**, and **order**.

Table 5-99 Parameter description

Parameter	Description
name	Cluster name.
status	Status code of a cluster. For details, see **Status Code**.
cursor	Cluster ID.

Parameter	Description
next	Whether to switch to the previous or next page. The value can be **true** or **false**. The default value is **true**.
limit	Number of returned data records. The default value is **10**.
order	Query result display. Value **asc** indicates that the query results are displayed in ascending order, and value **desc** indicates that the query results are displayed in descending order.

URI

- URI format

  GET /v1.0/{project_id}/reserved_clusters
- Parameter description

  Table 5-100 URI parameter description

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

Request

None

Response

- Example response

```
{
 "message_id": "CS.20040",
   "message": "Querying of the cluster list succeeds.",
 "payload": {
   "total": 1,
   "clusters": [
     {
       "cluster_id": 10,
       "name": "mycluster2",
       "desc": "Cluster description",
       "status_code": 5,
       "status_name": "cluster_running",
       "status_desc": "cluster_running",
       "spu_used": 0,
       "maximum_spu_quota": 100,

       "region_id": "cn-north-1",
```

```
      "error_reason": "",
      "created_at": 1550081520879,
      "vpc_cidr": "",
      "subnet_cidr": "",
      "subnet_gateway": "",
      "manager_node_spu": "2",
      "sys_enterprise_project_id": "0",
      "sys_enterprise_project_name": ""
    }
   ]
 },
 "current_time": 1550199862985
}
```


- Parameter description

Table 5-101 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
payload	No	None	Information about the exclusive cluster.
total	No	Int	Number of clusters.
clusters	No	None	Cluster list.
cluster_id	No	Int	Cluster ID.
name	No	String	Cluster name.
desc	No	String	Cluster description.
status_code	No	Int	Status code for a cluster.
status_name	No	String	Status name of a cluster. Possible values are: • cluster_creating • cluster_create_fail • cluster_create_success • cluster_starting • cluster_running • cluster_deleting • cluster_delete_fail
status_desc	No	String	Status description of a cluster.
spu_used	No	Int	SPU quota used by a cluster.
maximum_spu_quota	No	Int	Maximum SPU quota of a cluster.

Parameter	Mandatory	Type	Description
region_id	No	String	Region where a cluster is currently located.
error_reason	No	String	Cause of the error that results in cluster status exceptions.
created_at	No	Int	Time when a cluster is created.
vpc_cidr	No	String	VPC CIDR block where the cluster is located.
subnet_cidr	No	String	Subnet CIDR block where the cluster is located.
subnet_gateway	No	String	Subnet gateway for the cluster.
manager_node_spu	No	Int	SPU used for the management nodes of the cluster.
sys_enterprise_project_id	No	String	ID of the enterprise project to which the cluster belongs.
sys_enterprise_project_name	No	String	Name of the enterprise project to which the cluster belongs.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-102 describes the status code.

Table 5-102 Status code

Status Code	Description
200	Querying the list of clusters of a tenant succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.5 Updating an Exclusive Cluster

Function

This API is used to update exclusive clusters created by a user.

URI

- URI format

  PATCH /v1.0/{project_id}/reserved_cluster/{cluster_id}

- Parameter description

  Table 5-103 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	ID of an exclusive cluster.

Request

Submit the request for modifying an exclusive cluster.

- Example request

```
{
"name": "my_cluster",
  "desc": "This is my_cluster.",
  "maximum_spu_quota": 100
}
```

- Parameter description

  Table 5-104 Request parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Cluster name.
desc	No	String	Cluster description.
maximum_spu_quota	No	Int	Maximum SPU quota of a cluster.
tags	No	String	Job tag.
sys_enterprise_project_id	No	String	ID of the enterprise project to which a job belongs.
manager_node_spu	No	String	SPU specifications of management nodes.

Response

- Example response

```
{
  "message_id": "CS.21030",
```

```
  "message": "Cluster 1000000 is updated successfully.",
  "current_time": 1550201682465
}
```

- Parameter description

Table 5-105 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-106 describes the status code.

Table 5-106 Status code

Status Code	Description
200	An exclusive cluster is successfully updated.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.6 Terminating an Exclusive Cluster

Function

This API is used to terminate an exclusive cluster created by a user.

URI

- URI format

  POST /v1.0/{project_id}/reserved_cluster/{cluster_id}/stop
- Parameter description

Table 5-107 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	ID of an exclusive cluster.

Request

- Example request

```
{
   "cluster_id":"10"
}
```

- Parameter description

Table 5-108 Request parameter

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	ID of an exclusive cluster.

Response

- Example response

```
{
 "message_id": "CS.20050",
  "message": "Cluster is stopping.",
  "current_time": 1533686888000
}
```

- Parameter description

Table 5-109 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status code

Table 5-110 describes the status code.

Table 5-110 Status code

Status Code	Description
200	The request for terminating an exclusive cluster is delivered successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.7 Restarting an Exclusive Cluster

Function

This API is used to restart an exclusive cluster created by a user.

URI

- URI format
  POST /v1.0/{project_id}/reserved_cluster/{cluster_id}/restart
- Parameter description

  Table 5-111 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	ID of an exclusive cluster.

Request

- Example request

```
{
   "cluster_id":"10"
}
```

- Parameter description

  Table 5-112 Request parameter

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	ID of an exclusive cluster.

Response

- Example response

```
{
"message_id": "CS.20049",
  "message": "Cluster is restarting.",
  "current_time": 1533686888000
}
```

- Parameter description

Table 5-113 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-114 describes the status code.

Table 5-114 Status code

Status Code	Description
200	The request for restarting an exclusive cluster is delivered successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.8 Querying the List of Jobs Running in an Exclusive Cluster

Function

This API is used to query the list of jobs running in an exclusive cluster. The following parameters are involved in this API: **name**, **status**, **show_detail**, **cursor**, **next**, **limit**, and **order**.

URI

- URI format

  GET /v1.0/{project_id}/reserved_cluster/{cluster_id}/jobs

- Parameter description

Table 5-115 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Integer	ID of an exclusive cluster.
name	No	String	Job name.
status	No	String	Job status code. The options are as follows: • job_init • job_submitting • job_submit_fail • job_running • job_running_exception • job_canceling • job_cancel_success • job_cancel_fail
show_detail	No	Boolean	Whether to return job details. The default value is **false**.
cursor	No	Long	Job ID.
next	No	Boolean	Whether to switch to the next page. The default value is **true**.
limit	No	Integer	Number of returned data records. The default value is **20**.
order	No	String	Query result display. Value **asc** indicates that the query results are displayed in ascending order, and value **desc** indicates that the query results are displayed in descending order. The default value is **desc**.

Request

None

Response

- Example response

```
{
 "message_id": "CS.23000",
 "message": "Querying of the job list succeeds.",
 "payload": {
  "total": 3,
  "jobs": [
   {
    "job_id": 52912,
    "name": "carbondata0809",
    "desc": "",
    "user_name": "cs_testuser",
    "job_type": "flink_sql_job",
    "status": "job_running",
    "status_desc": "",
    "create_time": 1533803494092,
    "start_time": 1533805889940,
    "duration": 207119,
    "root_id": -1,
    "graph_editor_enabled": false
   },
   {
    "job_id": 52819,
    "name": "testSparkJar",
    "desc": "",
    "user_name": "cs_testuser",
    "job_type": "spark_streaming_jar_job",
    "status": "job_running",
    "status_desc": "",
    "create_time": 1533798894629,
    "start_time": 1533799258793,
    "duration": 6838266,
    "root_id": -1,
    "graph_editor_enabled": false
   },
   {
    "job_id": 52818,
    "name": "testFlinkJar",
    "desc": "testFlinkJar",
    "user_name": "cs_testuser",
    "job_type": "flink_jar_job",
    "status": "job_running",
    "status_desc": "",
    "create_time": 1533798799737,
    "start_time": 1533798894283,
    "duration": 7202776,
    "root_id": -1,
    "graph_editor_enabled": false
   }
  ]
 },
 "current_time": 1533806097155
}
```


- Parameter description

Table 5-116 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Parameter	Mandatory	Type	Description
payload	No	None	Information about the list of jobs running in an exclusive cluster.
total	No	Int	Number of records in the query result.
jobs	No	None	Information about a job.
job_id	Yes	Long	Job ID.
name	No	String	Job name.
desc	No	String	Job description.
username	No	String	Username. This parameter is valid only when **show_detail** is set to **true**.
job_type	No	String	Job type.
status	No	String	Job status.
status_desc	No	String	Description of job status.
create_time	Yes	Int	Time when a job is created.
start_time	No	Int	Time when a job is enabled. Value **0** indicates that the job has not been enabled.
duration	No	Int	Running duration of a job, expressed by milliseconds.
user_id	No	String	ID of the user who creates the job. This parameter is valid only when **show_detail** is set to **true**.
cluster_id	No	Int	Resource ID of an exclusive cluster. Ensure that the current user has been authorized to use resources of the exclusive cluster.
project_id	No	String	ID of the project to which a job belongs. This parameter is valid only when **show_detail** is set to **true**.
sql_body	No	String	Stream SQL statement.
run_mode	No	String	Running mode of a job. A job can run in shared or exclusive mode.
spu_number	No	Int	Number of SPUs selected for a job.
parallel_number	No	Int	Number of parallel jobs set by a user.

Parameter	Mandatory	Type	Description
job_config	No	None	Job configuration. This parameter is valid only when **show_detail** is set to **true**.
checkpoint_enabled	No	boolean	Whether to enable the automatic job snapshot function. • **true**: indicates to enable the automatic job snapshot function. • **false**: indicates to disable the automatic job snapshot function. • Default value: **false**
checkpoint_mode	No	String	Snapshot mode. The values include: • ExactlyOnce • AtLeastOnce
checkpoint_interval	No	Int	Snapshot interval, expressed by seconds.
log_enabled	No	boolean	Whether to enable the log storage function.
obs_bucket	No	String	Name of an OBS bucket.
checkpoint_interval	No	Int	Snapshot interval, expressed by seconds.
jar_url	No	String	OBS path of the JAR package. This parameter is valid only when **show_detail** is set to **true**.
main_class	No	String	OBS path where users are authorized to save the snapshot when **checkpoint_enabled** is set to **true**. This parameter is valid only when **show_detail** is set to **true**.
args	No	String	Job running parameter of the JAR package. This parameter is valid only when **show_detail** is set to **true**.
execution_graph	No	String	Job execution plan. This parameter is valid only when **show_detail** is set to **true**.
update_time	No	Int	Job update time. This parameter is valid only when **show_detail** is set to **true**.

Status Code

Table 5-117 describes the status code.

Table 5-117 Status code

Status Code	Description
200	Querying the list of jobs running in an exclusive cluster succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.9 Add an IP-Domain Mapping

Function

This API is used to add an IP-domain mapping for a specified cluster.

URI

- URI format

  POST /v1.0/{project_id}/reserved_cluster/{cluster_id}/hosts

- Parameter description

  Table 5-118 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.

Request

- Example request

```
{
  "hosts_info": [
    {
      "hostname": "test.cs.com",
      "ip": "10.113.27.138"
    }
  ]
}
```

- Parameter description

Table 5-119 Request parameters

Parameter	Mandatory	Type	Description
hosts_info	Yes	Array	Information about the IP-domain mapping to be added.
hostname	Yes	String	Domain name to be added.
ip	Yes	String	IP address to be added.

Response

- Example response

```
[
 {
  "message_id": "CS.22018",
  "message": "Host information is added successfully.",
  "payload": {
   "host_id": 5
  },
  "current_time": 1550210121485
 }
]
```

- Parameter description

Table 5-120 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
host_id	No	Long	Domain name ID.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-121 describes the status code.

Table 5-121 Status code

Status Code	Description
200	The IP-domain mapping is added successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.10 Updating the IP-Domain Mapping

Function

This API is used to update a specified IP-domain mapping.

URI

- URI format

  PATCH /v1.0/{project_id}/reserved_cluster/{cluster_id}/hosts/{host_id}

- Parameter description

  Table 5-122 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.
host_id	Yes	Domain name ID.

Request

- Example request

```
{
  "hosts_info": {
    "hostname": "test.cs.com",
    "ip": "10.113.27.139"
  }
}
```

- Parameter description

  Table 5-123 Request parameters

Parameter	Mandatory	Type	Description
hosts_info	Yes	None	Information about the new IP address and domain name.
hostname	Yes	String	New domain name.
ip	Yes	String	New IP address.

Response

- Example response

```
{
  "message_id":"CS.22019",
  "message":"Host information is updated successfully.",
  "current_time":1533686888000
}
```

- Parameter description

Table 5-124 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-125 describes the status code.

Table 5-125 Status Code

Status Code	Description
200	The IP-domain mapping is updated successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.11 Querying an IP-Domain Mapping

Function

This API is used to query information about the IP-domain mapping in a specified cluster.

The following parameters are involved in this API: **query_string**, **cursor**, **next**, **limit**, and **order**.

Table 5-126 Parameter description

Parameter	Description
query_string	IP address or domain name to be queried. Fuzzy match is supported.
cursor	Domain name ID.
next	Whether to switch to the next page.
limit	Number of returned data records.
order	Sorting style of the query results. Query results can be sorted in ascending or descending (default) order.

URI

- URI format

  GET /v1.0/{project_id}/reserved_cluster/{cluster_id}/hosts

- Parameter description

  Table 5-127 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.

Request

None

Response

- Example response

```
{
 "message_id": "CS.22017",
  "message": "Querying of the host list succeeds.",
 "payload": {
  "hosts_info": [{
    "id": 3,
    "info": {
      "hostname": "test.cs.com",
      "ip": "10.113.27.139"
    }
  }],
 "total":1
 },
 "current_time": 1533686888000
}
```

- Parameter description

Table 5-128 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	Information about the IP-domain mapping of the cluster.
hosts_info	No	None	Details about the IP-domain mapping of the cluster.
id	No	Long	Domain name ID.
info	No	None	Information about the IP-domain mapping.
hostname	No	String	Domain name.
ip	No	String	IP address.
total	No	Int	Total number of IP-domain mappings.

Status Code

Table 5-129 describes the status code.

Table 5-129 Status Code

Status Code	Description
200	The query of information about the IP-domain mapping of the cluster succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.12 Deleting an IP-Domain Mapping

Function

This API is used to delete the IP-domain mapping.

URI

- URI format

  DELETE /v1.0/{project_id}/reserved_cluster/{cluster_id}/hosts

- Parameter description

  Table 5-130 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.

Request

- Example request

```
{
  "hosts_id": [3]
}
```

- Parameter description

  Table 5-131 Request parameter

Parameter	Mandatory	Type	Description
hosts_id	Yes	Array[Long]	Domain name ID array.

Response

- Example response

```
[
  {
    "message_id": "CS.22020",
    "message": "Host information is deleted successfully.",
    "current_time": 1533686888000
  }
]
```

- Parameter description

Table 5-132 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-133 describes the status code.

Table 5-133 Status Code

Status Code	Description
200	The IP-domain mapping is deleted successfully.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.13 Adding the hosts File

Function

This API is used to add an IP-domain mapping for a specified cluster through file uploading. The **hosts_file** parameter is involved in this API, and the parameter must be in the "form-data" format.

Table 5-134 Parameter description

Parameter	Description
hosts_file	**hosts** file to be uploaded.

URI

- URI format

  POST /v1.0/{project_id}/reserved_cluster/{cluster_id}/hosts_file
- Parameter description

Table 5-135 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.

Request

The request format is form-data. For details about the parameter configuration, see **Table 5-134**.

Response

- Example response

```
[
 {
  "message_id":"CS.22018",
  "message":"Host information is added successfully.",
  "current_time":1533686888000,
  "payload":{
   "host_id":1
  }
 }
]
```

- Parameter description

Table 5-136 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
host_id	No	Long	Domain name ID.

Status Code

Table 5-137 describes the status code.

Table 5-137 Status code

Status Code	Description
200	The IP-domain mapping is added successfully.

Status Code	Description
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.14 Obtaining User Quota Information About a Tenant

Function

This API allows tenants to query user quota information. The following parameters are involved in this API: **name**, **order**, **cursor**, **next**, and **limit**.

Table 5-138 Parameter description

Parameter	Description
name	Cluster name.
cursor	User ID.
next	Whether to switch to the previous or next page.
limit	Number of returned data records.
order	Query result display. Value **asc** indicates that the query results are displayed in ascending order, and value **desc** indicates that the query results are displayed in descending order.

URI

- URI format

  GET /v1.0/{project_id}/user_quotas
- Parameter description

  Table 5-139 URI parameter description

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

Request

None

Response

- Example response

```
{
 "message_id": "CS.21093",
 "message": "Querying of the user quota list succeeds.",
 "payload": {
  "total": 2,
  "users": [
   {
    "user_id": "ac4eaa303639409c8ab099d55eb1538e",
    "user_name": "cs-testuser1",
    "spu_used": 0,
    "spu_quota": 1000,
    "clusters": []
   },
   {
    "user_id": "5138c47cbd3a408ebc997b7791aab411",
    "user_name": "cs-testuser",
    "spu_used": 6,
    "spu_quota": 1000,
    "clusters": [
     {
      "id": 1002703,
      "name": "opsadmtest_cluster"
     },
     {
      "id": 1002684,
      "name": "cs_standalone_tsq"
     },
     {
      "id": 1002640,
      "name": "test-wt"
     },
     {
      "id": 1002444,
      "name": "test-issue-nodel"
     }
    ]
   }
  ],
 "current_time": 1533867744575
 }
}
```

- Parameter description

Table 5-140 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	User quota information.

Parameter	Mandatory	Type	Description
total	No	Int	Number of user quota query results.
users	No	Int	User quota information.
user_id	No	String	User ID.
user_name	No	String	Username.
spu_used	No	Int	SPU quota used by a user.
spu_quota	No	Int	Maximum SPU quota of a user.
clusters	No	None	User cluster.
id	No	Int	Cluster ID.
name	No	String	Cluster name.

Status Code

Table 5-141 describes the status code.

Table 5-141 Status code

Status Code	Description
200	Querying user quota information about a tenant succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.15 Querying Quota Information About a Specified User

Function

This API is used to query quota information about a specified user.

URI

- URI format

  GET /v1.0/{project_id}/user_quota/{user_id}
- Parameter description

Table 5-142 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
user_id	Yes	User ID.

Request

None

Response

- Example response

```
{
"message_id": "CS.10001",
  "message": "Querying of the user quota succeeds.",
  "current_time": 1533686888000,
  "payload": {
    "user_id": "ac4eaa303639409c8ab099d55eb1538e",
    "user_name": "cs_testuser",
    "spu_used": 20,
    "spu_quota": 100,
    "clusters": [
      {
        "id": 1000000,
        "name": "cluster_001"
      }
    ]
  }
}
```

- Parameter description

Table 5-143 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.
payload	No	None	User quota information.
user_id	No	String	User ID.
user_name	No	String	Username.
spu_used	No	String	SPU quota used by a user.
spu_quota	No	Int	Maximum SPU quota of a user.

Parameter	Mandatory	Type	Description
clusters	No	None	User cluster.
id	No	Int	Cluster ID.
name	No	String	Cluster name.

Status Code

Table 5-144 describes the status code.

Table 5-144 Status code

Status Code	Description
200	Querying quota information about a specified user succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.16 Updating Quota Information About a Specified User

Function

This API allows tenants to modify quota information about a specified user as needed.

URI

- URI format
  PATCH /v1.0/{project_id}/user_quota/{user_id}
- Parameter description

Table 5-145 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
user_id	Yes	User ID.

Request

Update quota information about a specified user. The settings of related request parameters are in JSON format.

- Example request

```
{
"spu_quota": 100,
 "clusters": [
  111,
  112
 ]
}
```

- Parameter description

Table 5-146 Request parameters

Parameter	Mandatory	Type	Description
spu_quota	No	Int	Maximum SPU quota of a user.
clusters	No	Int	Maximum SPU quota of a cluster.

Response

- Example response

```
{
"message_id": "CS.10001",
 "message": "Updating the user quota succeeds.",
 "current_time": 1533686888000
}
```

- Parameter description

Table 5-147 Response parameters

Parameter	Mandatory	Type	Description
message_id	No	String	Message type ID.
message	No	String	Message content.
current_time	No	Int	Current time, expressed by milliseconds.

Status Code

Table 5-148 describes the status code.

Table 5-148 Status Code

Status Code	Description
200	Updating quota information about a specified user succeeds.
400	The input parameters are invalid.

Error Code

For details, see **Error Codes**.

5.3.17 Creating a VPC Peering Connection

Function

This API is used to create a VPC peering connection used to connect to other VPCs.

URI

- URI format

  POST /v1.0/{project_id}/reserved_cluster/{cluster_id}/peering
- Parameter description

  Table 5-149 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.

Request

Create a VPC peering connection used to connect to other VPCs. The request parameter setting is in JSON format.

- Example request

```
{
  "vpc_id": "f4da5a55-fcd9-4a93-8b86-5b81585e14bb",
  "name": "myVpc"
}
```

- Parameter description

Table 5-150 Request parameters

Parameter	Mandatory	Type	Description
vpc_id	Yes	String	ID of the peer VPC.
name	Yes	String	Name of the VPC peering connection.
accept_tenant_id	No	String	ID of the tenant to which the peer VPC belongs. This parameter is required only when a VPC peering connection to the VPC of another tenant is created.

Response

- Example response

```
{
  "peering":{
    "name":"myVpc",
    "id":"7e974044-bcc1-4f25-aff8-5f10cb347b0f",
    "request_vpc_info":{
      "vpc_id":"83081290-198e-48e2-8e2d-64c50023d136",
      "tenant_id":"8185f87d043f4519a68420a282b5e2fa"
    },
    "accept_vpc_info":{
      "vpc_id":"f4da5a55-fcd9-4a93-8b86-5b81585e14bb",
      "tenant_id":"4f30b9215e054c908d18f2b150131646"
    },
    "status":"PENDING_ACCEPTANCE"
  }
}
```

- Parameter description

Table 5-151 Response parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the VPC peering connection.
id	Yes	String	ID of the VPC peering connection.
request_vpc_info	Yes	Struct	Information about the local VPC.
accept_vpc_info	Yes	Struct	Information about the peer VPC.
vpc_id	Yes	String	ID of the peer VPC.
tenant_id	Yes	String	ID of the tenant to which the peer VPC belongs.

Parameter	Mandatory	Type	Description
status	Yes	String	Status of the VPC peering connection. Available options are as follows: **PENDING_ACCEPTANCE**, **REJECTED**, **EXPIRED**, **DELETED**, and **ACTIVE**.

Status Code

Table 5-152 Status code

Status Code	Description
201	A VPC peering connection is created successfully.
400	The input parameter is invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

5.3.18 Querying the List of VPC Peering Connections

Function

This API is used to query the list of VPC peering connections.

URI

- URI format
  GET /v1.0/{project_id}/reserved_cluster/{cluster_id}/peering
- Parameter description

Table 5-153 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.

Request

None

Response

- Example response

```
{
  "peerings":[
    {
      "name":"myVpc",
      "id":"7e974044-bcc1-4f25-aff8-5f10cb347b0f",
      "request_vpc_info":{
        "vpc_id":"83081290-198e-48e2-8e2d-64c50023d136",
        "tenant_id":"8185f87d043f4519a68420a282b5e2fa",
        "vpc_cidr":"172.16.0.0/23",
        "subnet_cidr":"172.16.0.0/23"
      },
      "accept_vpc_info":{
        "vpc_id":"f4da5a55-fcd9-4a93-8b86-5b81585e14bb",
        "tenant_id":"4f30b9215e054c908d18f2b150131646"
      },
      "status":"PENDING_ACCEPTANCE"
    }
  ]
}
```

- Parameter description

Table 5-154 Response parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the VPC peering connection.
id	Yes	String	ID of the VPC peering connection.
request_vpc_info	Yes	Struct	Information about the local VPC.
accept_vpc_info	Yes	Struct	Information about the peer VPC.
vpc_id	Yes	String	ID of the peer VPC.
tenant_id	Yes	String	ID of the tenant to which the peer VPC belongs.
status	Yes	String	Status of the VPC peering connection. Available options are as follows: **PENDING_ACCEPTANCE**, **REJECTED**, **EXPIRED**, **DELETED**, and **ACTIVE**.

Status Code

Table 5-155 Status code

Status Code	Description
200	Querying the list of VPC peering connections succeeds.
400	The output parameters are invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

5.3.19 Querying a VPC Peering Connection

Function

This API is used to query a specified VPC peering connection.

URI

- URI format
  GET /v1.0/{project_id}/reserved_cluster/{cluster_id}/peering/{peering_id}
- Parameter description

  Table 5-156 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.
peering_id	Yes	ID of the VPC peering connection.

Request

None

Response

- Example response

```
{
  "peering":{
```

```
    "name":"myVpc",
    "id":"7e974044-bcc1-4f25-aff8-5f10cb347b0f",
    "request_vpc_info":{
      "vpc_id":"83081290-198e-48e2-8e2d-64c50023d136",
      "tenant_id":"8185f87d043f4519a68420a282b5e2fa"
    },
    "accept_vpc_info":{
      "vpc_id":"f4da5a55-fcd9-4a93-8b86-5b81585e14bb",
      "tenant_id":"4f30b9215e054c908d18f2b150131646"
    },
    "status":"ACTIVE"
  }
}
```


- Parameter description

Table 5-157 Response parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the VPC peering connection.
id	Yes	String	ID of the VPC peering connection.
request_vpc_info	Yes	Struct	Information about the local VPC.
accept_vpc_info	Yes	Struct	Information about the peer VPC.
vpc_id	Yes	String	ID of the peer VPC.
tenant_id	Yes	String	ID of the tenant to which the peer VPC belongs.
status	Yes	String	Status of the VPC peering connection. Available options are as follows: **PENDING_ACCEPTANCE**, **REJECTED**, **EXPIRED**, **DELETED**, and **ACTIVE**.

Status Code

Table 5-158 Status code

Status Code	Description
200	Querying the VPC peering connection succeeds.
400	The input parameter is invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

5.3.20 Deleting a VPC Peering Connection

Function

This API is used to delete a specified VPC peering connection.

URI

- URI format
  DELETE /v1.0/{project_id}/reserved_cluster/{cluster_id}/peering/{peering_id}
- Parameter description

Table 5-159 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.
peering_id	Yes	ID of the VPC peering connection.

Request

None

Response

None

Status Code

Table 5-160 Status code

Status Code	Description
204	Deleting the VPC peering connection succeeds.
400	The input parameter is invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

5.3.21 Accepting a VPC Peering Connection Request

Function

This API is used to accept a VPC peering connection request. Use the following method to send a request to the endpoint of the peer VPC:

PUT https://***VPC endpoint***/v2.0/vpc/peerings/{peering_id}/accept

For details about the VPC endpoint, see **Regions and Endpoints**.

URI

- URI format

  PUT /v2.0/vpc/peerings/{peering_id}/accept
- Parameter description

  Table 5-161 URI parameter description

Parameter	Mandatory	Description
peering_id	Yes	ID of the VPC peering connection.

Request

None

Response

- Example response

```
{
  "status": "ACTIVE",
  "name": "myVpcaa",
  "tenant_id": "2a62704287e74bdb90dc00ff4235748c",
  "request_vpc_info": {
    "vpc_id": "ecde182e-b7fa-4a8e-85f6-0db8bfa2583a",
    "tenant_id": "b17e103dab334f18b74335449d3eb763"
  },
  "accept_vpc_info": {
    "vpc_id": "032d0cd5-4121-4b6c-9170-0dc4337173f6",
    "tenant_id": "2a62704287e74bdb90dc00ff4235748c"
  },
  "id": "0a5a0dae-39f2-44a1-a6b7-bcc3383865a3"
}
```

- Parameter description

Table 5-162 Response parameters

Parameter	Mandatory	Type	Description
status	Yes	String	Status of the request for accepting the VPC peering connection.
name	Yes	String	Name of the VPC peering connection.
tenant_id	Yes	String	Tenant ID.
vpc_id	Yes	String	ID of the VPC for which a route is to be added.
tenant_id	Yes	String	ID of the tenant who requests to add a route.
id	Yes	String	ID of a route.

Status Code

Table 5-163 Status code

Status Code	Description
200	The request of accepting the VPC peering connection succeeds.
400	The input parameter is invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

5.3.22 Creating a Route to the Local VPC

Function

This API is used to create a route to the local end of a VPC peering connection. The prerequisite for calling this API is that the VPC peering connection request has been accepted. For details, see **Accepting a VPC Peering Connection Request**.

URI

- URI format

POST /v1.0/{project_id}/reserved_cluster/{cluster_id}/peering/{peering_id}/route

- Parameter description

Table 5-164 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.
peering_id	Yes	ID of the VPC peering connection.

Request

- Example request

```
{
  "destination": "192.168.0.0/16"
}
```

- Parameter description

Table 5-165 Request parameter

Parameter	Mandatory	Type	Description
destination	Yes	String	Destination address in the CIDR notation format, which is generally the CIDR of the peer VPC.

Response

- Example response

```
{
   "route":{
      "type":"peering",
      "nexthop":"7e974044-bcc1-4f25-aff8-5f10cb347b0f",
      "destination":"192.168.0.0/16",
      "vpc_id":"83081290-198e-48e2-8e2d-64c50023d136",
      "tenant_id":"8185f87d043f4519a68420a282b5e2fa",
      "id":"75d289f7-9022-4950-a137-fed7aeae9cb1"
   }
}
```

- Parameter description

Table 5-166 Response parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Route type. Generally, this parameter is set to **peering**.
nexthop	Yes	String	Next hop address of a route. Generally, this parameter is set to the ID of the VPC peering connection.
destination	Yes	String	Destination address in the CIDR notation format.
vpc_id	Yes	String	ID of the VPC for which a route is to be added.
tenant_id	Yes	String	ID of the tenant who requests to add a route.
id	Yes	String	ID of a route.

Status Code

Table 5-167 Status code

Status Code	Description
201	A route is created successfully.
400	The input parameter is invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

5.3.23 Creating a Route to the Peer VPC

Function

You need to create the routes for both the receiver (local VPC) and the sender (peer VPC). For details about how to create a local route, see **Creating a Route to the Local VPC**.

The prerequisite for calling this API is that the VPC peering connection request has been accepted. For details, see **Accepting a VPC Peering Connection Request**.

Use the following method to send a request to the endpoint of the peer VPC:

POST https://VPC endpoint/v2.0/vpc/routes

URI

POST /v2.0/vpc/routes

Request

- Example request

```
POSThttps: //{Endpoint}/v2.0/vpc/routes
{
   "route": {
      "type": "peering",
      "nexthop": "60c809cb-6731-45d0-ace8-3bf5626421a9",
      "destination": "192.168.200.0/24",
      "vpc_id": "ab78be2d-782f-42a5-aa72-35879f6890ff"
   }
}
```

- Parameter description

Table 5-168 Request parameter

Parameter	Mandatory	Description
route	Yes	Route object list. For details, see **Table 5-169**.

Table 5-169 route objects

Attribute	Type	Mandatory	Description
destination	String	Yes	Destination address in the CIDR notation format, for example, 192.168.200.0/24.
nexthop	String	Yes	Next hop. If the route type is **peering**, enter the VPC peering connection ID.
type	String	Yes	Route type. Currently, the value can only be **peering**.
vpc_id	String	Yes	ID of the VPC requesting for creating a route.

Response

- Example response

```
{
   "route": {
      "type": "peering",
      "nexthop": "60c809cb-6731-45d0-ace8-3bf5626421a9",
      "destination": "192.168.200.0/24",
      "vpc_id": "ab78be2d-782f-42a5-aa72-35879f6890ff",
      "tenant_id": "6fbe9263116a4b68818cf1edce16bc4f",
      "id": "3d42a0d4-a980-4613-ae76-a2cddecff054"
```



```
  }
}
```

- Parameter description

Table 5-170

Parameter	Type	Description
route	Object	Route object list. For details, see **Table 4**.

Table 5-171 Response parameters

Attribute	Type	Description
id	String	ID of a route.
destination	String	Destination address in the CIDR notation format, for example, 192.168.200.0/24.
nexthop	String	Next hop. If the route type is **peering**, enter the VPC peering connection ID.
type	String	Route type. Currently, the value can only be **peering**.
vpc_id	String	VPC of the route. Set this parameter to the existing VPC ID.
tenant_id	String	**Project ID**.

Status Code

Table 5-172 Status codes

Status Code	Description
201	A route is created successfully.
400	The input parameter is invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

5.3.24 Query the Route List

Function

This API is used to query the list of routes of a VPC peering connection.

URI

- URI format

  GET /v1.0/{project_id}/reserved_cluster/{cluster_id}/peering/{peering_id}/route

- Parameter description

  Table 5-173 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.
peering_id	Yes	ID of the VPC peering connection.

Request

None

Response

- Example response

```
{
  "routes":[{
    "type":"peering",
    "nexthop":"7e974044-bcc1-4f25-aff8-5f10cb347b0f",
    "destination":"192.168.0.0/16",
    "vpc_id":"83081290-198e-48e2-8e2d-64c50023d136",
    "tenant_id":"8185f87d043f4519a68420a282b5e2fa",
    "id":"75d289f7-9022-4950-a137-fed7aeae9cb1"
  }]
}
```

- Parameter description

  Table 5-174 Response parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Route type. Generally, this parameter is set to **peering**.

Parameter	Mandatory	Type	Description
nexthop	Yes	String	Next hop address of a route. Generally, this parameter is set to the ID of the VPC peering connection.
destination	Yes	String	Destination IP address or CIDR block.
vpc_id	Yes	String	ID of the VPC for which a route is to be added.
tenant_id	Yes	String	ID of the tenant who requests to add a route.
id	Yes	String	ID of a route.

Status Code

Table 5-175 Status Code

Status Code	Description
200	Route list query succeeds.
400	The input parameter is invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

5.3.25 Deleting a Route

Function

This API is used to delete a specific route.

URI

- URI format

  DELETE /v1.0/{project_id}/reserved_cluster/{cluster_id}/peering/{peering_id}/route/{route_id}

- Parameter description

Table 5-176 URI parameter description

Parameter	Mandatory	Description
project_id	Yes	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	Cluster ID.
peering_id	Yes	ID of the VPC peering connection.
route_id	Yes	ID of a route.

Request

None

Response

None

Status Code

Table 5-177 Status code

Status Code	Description
204	Failed to delete the route.
400	The input parameter is invalid.
500	Failed to complete the request because of an internal service error.

Error Code

For details, see **Error Codes**.

6 Permissions Policies and Supported Actions

This chapter describes fine-grained permissions management for your CS. If your HUAWEI CLOUD account does not require individual IAM users, you can skip this chapter.

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. For more information about policy syntax and example policies, see **Permissions Management**.

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. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant CS users only the permissions for managing a certain type of cloud servers.

NOTE

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

An account has all of the permissions required to call all APIs, but IAM users must have the required permissions specifically assigned. The permissions required for calling an API are determined by the actions supported by the API. Only users that have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user queries the list of CS jobs using an API, the user must have been granted permissions that allow the **cs:jobs:list** action.

Supported Actions

Actions supported by policies are specific to APIs. The following are common concepts related to policies:

- Permissions: Defined by actions in a custom policy.
- Actions: Added to a custom policy to control permissions for specific operations.
- Authorization Scope: A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions supporting both IAM and enterprise projects can be assigned to user groups and take effect in both IAM and Enterprise Management. Policies that only contain actions supporting IAM projects can be assigned to user groups and only take effect for IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Project. For details about the differences between IAM and enterprise projects, see **What Are the Differences Between IAM and Enterprise Management?**.
- APIs: REST APIs that can be called in a custom policy.

CS supports the following actions that can be defined in custom policies:

Table 6-1 API actions

Permission	Action	Authorization Scope	API
Obtaining Details About a Job	cs:jobs:get	• Supported: Projects Enterprise projects • Not supported: none	GET /v1.0/{project_id}/job/{job_id}
Obtains the Job List	cs:jobs:list	• Supported: Projects Enterprise projects • Not supported: none	GET /v1.0/{project_id}/jobs
Creating a Job	cs:jobs:create	• Supported: Projects Enterprise projects • Not supported: none	Creating a SQL job: POST /v1.0/{project_id}/sql_job Creating a user-defined job: POST /v1.0/{project_id}/jar_job

Permission	Action	Authorization Scope	API
Updating a Job	cs:jobs:update	• Supported: Projects Enterprise projects • Not supported: none	Updating a SQL job: PATCH /v1.0/{project_id}/sql_job Updating a user-defined job: PATCH /v1.0/{project_id}/jar_job
Deleting a Job	cs:jobs:delete	• Supported: Projects Enterprise projects • Not supported: none	DELETE /v1.0/{project_id}/job/{job_id}
Starting a Job	cs:jobs:start	• Supported: Projects Enterprise projects • Not supported: none	POST /v1.0/{project_id}/job/{job_id}/run
Stopping a Job	cs:jobs:stop	• Supported: Projects Enterprise projects • Not supported: none	POST /v1.0/{project_id}/job/{job_id}/stop
Obtaining Details About a Cluster	cs:clusters:get	• Supported: Projects Enterprise projects • Not supported: none	GET /v1.0/{project_id}/reserved_cluster/{cluster_id}
Obtaining the Cluster List	cs:clusters:list	• Supported: Projects Enterprise projects • Not supported: none	GET /v1.0/{project_id}/reserved_clusters
Creating a Cluster	cs:clusters:create	• Supported: Projects Enterprise projects • Not supported: none	POST /v1.0/{project_id}/reserved_cluster

Permission	Action	Authorization Scope	API
Updating a Cluster	cs:clusters:update	• Supported: Projects Enterprise projects • Not supported: none	PATCH /v1.0/{project_id}/reserved_cluster/{cluster_id}
Stopping a Cluster	cs:clusters:stop	• Supported: Projects Enterprise projects • Not supported: none	POST /v1.0/{project_id}/reserved_cluster/{cluster_id}/stop
Restarting a Cluster	cs:clusters:restart	• Supported: Projects Enterprise projects • Not supported: none	POST /v1.0/{project_id}/reserved_cluster/{cluster_id}/restart
Deleting a Cluster	cs:clusters:delete	• Supported: Projects Enterprise projects • Not supported: none	DELETE /v1.0/{project_id}/reserved_cluster/{cluster_id}
Querying Quota Information About a User	cs:clusterQuota:get	• Supported: Projects Enterprise projects • Not supported: none	GET /v1.0/{project_id}/user_quota/{user_id}
Querying Quota Information About all Users	cs:clusterQuota:list	• Supported: Projects Enterprise projects • Not supported: none	GET /v1.0/{project_id}/user_quotas
Modifying User Quotas	cs:clusterQuota:modify	• Supported: Projects Enterprise projects • Not supported: none	PATCH /v1.0/{project_id}/user_quota/{user_id}
Obtains Auditing Information	cs:auditLog:get	• Supported: Projects Enterprise projects • Not supported: none	GET /v1.0/{project_id}/audit_logs?resource_type=job

Permission	Action	Authorization Scope	API
Obtaining the Template List	cs:template:list	• Supported: Projects Enterprise projects • Not supported: none	GET /v1.0/{project_id}/job_templates
Creating a Template	cs:template:create	• Supported: Projects Enterprise projects • Not supported: none	POST /v1.0/{project_id}/job_template
Updating Information About a Template	cs:template:update	• Supported: Projects Enterprise projects • Not supported: none	PATCH /v1.0/{project_id}/job_template
Deleting a Template	cs:template:delete	• Supported: Projects Enterprise projects • Not supported: none	DELETE /v1.0/{project_id}/job_template/{template_id}

7 Appendix

7.1 Status Code

Table 7-1 describes the status code.

Table 7-1 Status code

Status Code	Code	Description
100	Continue	The client continues sending the request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server.
101	Switching Protocols	Switching protocols. The target protocol must be more advanced than the source protocol. For example, the current HTTP protocol is switched to a later version.
200	OK	The server has successfully processed the request.
201	Created	The request for creating a resource has been fulfilled.
202	Accepted	The request has been accepted, but the processing has not been completed.
203	Non-Authoritative Information	The server successfully processed the request, but is returning information that may be from another source.

Status Code	Code	Description
204	NoContent	The server has successfully processed the request, but has not returned any content. The status code is returned in response to an HTTP OPTIONS request.
205	Reset Content	The server has fulfilled the request, but the requester is required to reset the content.
206	Partial Content	The server has successfully processed a part of the GET request.
300	Multiple Choices	There are multiple options for the location of the requested resource. The response contains a list of resource characteristics and addresses from which the user or user agent (such as a browser) can choose the most appropriate one.
301	Moved Permanently	The requested resource has been assigned a new permanent URI, and the new URI is contained in the response.
302	Found	The requested resource resides temporarily under a different URI.
303	See Other	Retrieve a location. The response to the request can be found under a different URI and should be retrieved using a GET or POST method.
304	Not Modified	The requested resource has not been modified. When the server returns this status code, it does not return any resources.
305	Use Proxy	The requested resource must be accessed through a proxy.
306	Unused	The HTTP status code is no longer used.
400	BadRequest	Invalid request. The client should not repeat the request without modifications.
401	Unauthorized	The status code is returned after the client provides the authentication information, indicating that the authentication information is incorrect or invalid.
402	Payment Required	This status code is reserved for future use.

Status Code	Code	Description
403	Forbidden	The server understood the request, but is refusing to fulfill it. The client should not repeat the request without modifications.
404	NotFound	The requested resource cannot be found. The client should not repeat the request without modifications.
405	MethodNotAllowed	The method specified in the request is not supported for the requested resource. The client should not repeat the request without modifications.
406	Not Acceptable	The server cannot fulfill the request according to the content characteristics of the request.
407	Proxy Authentication Required	This status code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Time-out	The request timed out. The client may repeat the request without modifications at any later time.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This status code indicates that the resource that the client attempts to create already exits, or the request fails to be processed because of the update of the conflict request.
410	Gone	The requested resource is no longer available. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server refuses to process the request without a defined Content-Length.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.
413	Request Entity Too Large	The request is larger than that a server is able to process. The server may close the connection to prevent the client from continuing the request. If the server cannot process the request temporarily, the response will contain a Retry-After header field.

Status Code	Code	Description
414	Request-URI Too Large	The URI provided was too long for the server to process.
415	Unsupported Media Type	The server is unable to process the media format in the request.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the Expect request-header field.
422	UnprocessableEnti-ty	The request is well-formed but is unable to be processed due to semantic errors.
429	TooManyRequests	The client has sent more requests than its rate limit is allowed within a given amount of time, or the server has received more requests than it is able to process within a given amount of time. In this case, it is advisable for the client to re-initiate requests after the time specified in the Retry-After header of the response expires.
500	InternalServerError	The server is able to receive the request but it could not understand the request.
501	Not Implemented	The server does not support the requested function.
502	Bad Gateway	The server is acting as a gateway or proxy and receives an invalid request from a remote server.
503	ServiceUnavailable	The requested service is invalid. The client should not repeat the request without modifications.
504	ServerTimeout	The request cannot be fulfilled within a given time. This status code is returned to the client only when the **Timeout** parameter is specified in the request.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

7.2 Error Codes

If an error occurs in API calling, no result is returned. Identify the error cause based on the error codes of each API. If an error occurs in API calling, an HTTP

status code is returned. The response body contains the specific error code and information.

Example Value

```
{
    "error_id": "CS.11004",
    "error_name": "JOB_NAME_EXIST",
    "error_cause": "This job name is already existed",
    "error_suggestion": "Please choose a new job name",
    "error_time": 1538016904593
}
```

The following table describes the common error codes.

Table 7-2 Common error codes

Error Code	Status Code	Description	Solution
CS.10001	400	An internal error occurred on the service.	Submit a service ticket for troubleshooting or try again later.
CS.10002	400	The requested resource is forbidden.	Refer to the *Cloud Stream Service API Reference*.
CS.10003	400	The requested API is not supported. A complete API contains the API domain name, path, HTTP method, query parameters, request body, request header parameters, path parameters, response status code, response header, response body, and more.	Select a correct API and try again, or check whether the API specifications are met. For details, see the *Cloud Stream Service API Reference*.
CS.10004	400	The JSON format of the request body is incorrect.	Refer to the *Cloud Stream Service API Reference* or JSON format specifications.
CS.10005	400	You have no permission to perform this request.	Confirm your permissions.
CS.10006	400	The project ID in the URL is different from that in the IAM token.	Check whether the project ID in the IAM token is correct.
CS.10007	400	The IAM token has expired.	Log in again and re-call this API.
CS.10008	400	There is no token in the request header.	Add **X-Auth-Token** to the request header.
CS.10009	400	The account is frozen because the account balance is less than 0.	Top up the account.

Error Code	Status Code	Description	Solution
CS.10010	400	The user ID or project ID does not exist in the IAM token.	Contact technical support.
CS.10014	400	Permission denied or insufficient balance.	Grant the permission to the account or top up the account.
CS.10015	400	IAM request failed.	Try again later.
CS.10016	400	Real-time authentication is not performed before resource creation.	Complete the real-name authentication.
CS.11001	400	The maximum length of a SQL statement is 10,000 bytes.	Specify a shorter SQL statement.
CS.11002	400	The OBS bucket must be provided if the checkpoint function or OBS is enabled.	Configure **checkpoint_obs_bucket** in the request.
CS.11003	400	A maximum of 100 jobs in the **Draft** status are allowed.	Delete unused jobs in the **Draft** status.
CS.11004	400	The job name exists.	Select a new job name.
CS.11005	400	Failed to submit the stream SQL job.	Submit a service ticket for troubleshooting.
CS.11006	400	The maximum length of a job name is 57 bytes.	Specify a shorter job name.
CS.11007	400	The job name cannot be left blank.	Enter a job name.
CS.11008	400	Invalid **checkpoint_interval** setting.	As specified in the *Cloud Stream Service API Reference*, if **checkpointEnabled** is set to **true**, **checkpoint_interval** cannot be left blank.
CS.11009	400	Invalid **checkpoint_mode** setting.	As specified in the *Cloud Stream Service API Reference*, if **checkpointEnabled** is set to **true**, **checkpoint_mode** cannot be left blank.

Error Code	Status Code	Description	Solution
CS.11010	400	The **obs_bucket** parameter is not specified.	As specified in the *Cloud Stream Service API Reference*, if **checkpointEnabled** is set to **true**, **obs_bucket** cannot be left blank.
CS.11011	400	Invalid **obs_bucket** setting	As specified in the *Cloud Stream Service API Reference*, if **log_enabled** is set to **true**, **obs_bucket** cannot be left blank.
CS.11012	400	The SPU quantity for a job is beyond the value range.	Set the SPU quantity to a value ranging from 1 to 400.
CS.11013	400	The maximum number of parallel jobs is beyond the value range.	Set the maximum number of parallel jobs to a value ranging from 1 to 50.
CS.11014	400	The maximum length of a job description is 512 bytes.	Specify a shorter job description.
CS.11015	400	A job name consists of letters, digits, underscores (_), and hyphens (-).	Provide a valid job name.
CS.11016	400	Permission to access an OBS bucket has not been granted to CS.	Grant CS the permission to access this OBS bucket.
CS.11017	400	The uploaded file is not a JAR file.	Upload a JAR file.
CS.11018	400	The main class name of a JAR job includes a maximum of 512 bytes.	Specify a shorter main class name.
CS.11019	400	The main class arguments of a job include a maximum of 1,024 bytes.	Specify shorter main class arguments.
CS.11020	400	The job main class name includes spaces.	Delete the spaces in the job main class name.
CS.11021	400	The JAR file does not exist.	Select an available JAR file.
CS.11022	400	Parallelism of a job cannot exceed four times of the SPU quantity for the compute unit.	Set **Parallelism** to be not greater than four times of the SPU quantity for the compute unit.

Error Code	Status Code	Description	Solution
CS.11023	400	No edge computing group is specified.	Specify the **edge_group_id** parameter.
CS.11024	400	The job type and the running mode mismatch.	Use the correct running mode.
CS.11025	400	Invalid SPU quantity for the management unit.	Enter a valid number of SPUs. Ensure that the number of SPUs for management units is not less than 1 or not greater than the total number of SPUs minus 1 but does not exceed 4.
CS.11026	400	Parameters **executor_spu** and **executor_number** must be specified for Spark streaming JAR jobs. Total number of SPUs = executor_spu x executor_number + manager_spu	Provide a valid value.
CS.11027	400	The value of **Executor_spu** ranges from **1** to **4**.	Provide a valid value.
CS.11028	400	The value of **Executor_num** ranges from **1** to **100**.	Provide a valid value.
CS.11029	400	The parameter must be specified.	Configure the corresponding parameter.
CS.11030	400	The configuration file does not exist.	Select a valid configuration file.
CS.11031	400	The size of the uploaded local file cannot exceed 8 MB.	Upload files to OBS or upload a file not greater than 8 MB.
CS.11032	400	The SMN topic does not exist.	Select an existing SMN topic.
CS.11035	400	UDFs cannot be used in the shared cluster.	Select an exclusive cluster.
CS.11036	400	The idle state retention time ranges from 1s to 60s.	Provide a valid value.
CS.11037	400	Failed to aggregate job logs.	Try again later.

Error Code	Status Code	Description	Solution
CS.11038	400	The log dump function is disabled.	Enable the function on the job editing page.
CS.11039	400	Invalid dirty data policy.	Provide a valid dirty data policy.
CS.11040	400	Failed to aggregate user data.	Try again later.
CS.11041	400	Failed to upload user data.	Try again later.
CS.11098	400	Failed to obtain the APIG address of the job.	Try again later.
CS.12001	400	Invalid job ID. The job ID must be a digit greater than 0.	Provide a valid job ID, which can be obtained from the CS job list.
CS.12002	400	Failed to update the Stream SQL statements.	Check the system and try again.
CS.12003	400	Jobs in the **Running** or **Submitting** status cannot be updated.	Wait until the job is completed and try again.
CS.12004	400	The job does not exist.	Check the reason or create a job.
CS.12005	400	The job cannot be updated.	Check why the job cannot be updated.
CS.12006	400	The job name already exists.	Try a new job name.
CS.13001	400	Insufficient quota.	Contact the administrator to adjust the quota.
CS.13002	400	Insufficient SPUs.	Contact the administrator to adjust the cluster quota.
CS.13003	400	The snapshot or logging function has been enabled, but no OBS bucket is selected.	Select an OBS bucket or disable the snapshot or logging function.
CS.13004	400	Permission to access an OBS bucket has not been granted to CS.	Grant CS the permission to access the OBS bucket or disable the snapshot or logging function.
CS.13005	400	Incorrect SQL syntax.	View error details in the SQL editor and modify the statements.

Error Code	Status Code	Description	Solution
CS.13006	400	The job is running or is not allowed to run.	Check the job status.
CS.13007	400	The job ID does not exist.	Provide an available job ID, which can be obtained from the CS job list.
CS.13008	400	An exclusive cluster not in the **Running** status is selected for running jobs.	Provide an available cluster ID, which can be obtained from the CS cluster list.
CS.13009	400	User-defined jobs cannot run on the shared cluster.	Select an exclusive cluster.
CS.13010	400	Exceptions occur during job running.	Check job logs.
CS.13011	400	The user does not have the permission to use the cluster.	Select an available cluster.
CS.13012	400	Incorrect job type.	Enter a correct JAR job type.
CS.13013	400	The JAR file does not exist.	Upload a valid JAR file.
CS.13014	400	A job with Kafka as the source or sink stream cannot run on the shared cluster.	Select an exclusive cluster.
CS.13015	400	Incorrect type setting for the source or sink stream.	Enter a correct source/sink stream type.
CS.13016	400	A job with {0} as the source or sink stream cannot run on the shared cluster.	Select an exclusive cluster.
CS.13017	400	The job savepoint does not exist.	Create a job savepoint first. You can choose whether to create the savepoint when stopping a job.
CS.13018	400	No JAR file is contained in the path specified by **jar_url**.	Ensure that at least a valid JAR file is contained in the path specified by **jar_url**.
CS.13019	400	The **main_class** parameter must be specified when there are multiple JAR files in the path specified by **jar_url**.	Specify the **main_class** parameter when there are multiple JAR files in the path specified by **jar_url**.
CS.14001	400	The job execution graph does not exist.	Try again later.

Error Code	Status Code	Description	Solution
CS.15001	400	The job ID does not exist.	Provide an available job ID, which can be obtained from the CS job list.
CS.15002	400	You are not allowed to stop the job in the current status.	Refer to the *Cloud Stream Service API Reference*.
CS.16001	400	The job ID does not exist.	Provide an available job ID, which can be obtained from the CS job list.
CS.17001	400	The job ID does not exist.	Provide an available job ID, which can be obtained from the CS job list.
CS.18001	400	The maximum length of a SQL statement in a template is 10,000 bytes.	Specify a shorter SQL statement.
CS.18002	400	The template name already exists.	Select a new template name.
CS.18003	400	A maximum of 100 templates are allowed.	Delete unused templates.
CS.18004	400	The JSON format of the request body is incorrect.	Refer to the *Cloud Stream Service API Reference*.
CS.18005	400	The template does not exist.	Specify an existing template ID.
CS.18006	400	The template name cannot be left blank.	Specify a template name.
CS.18007	400	A template name consists of letters, digits, underscores (_), and hyphens (-).	Provide a valid template name.
CS.18008	400	The template description includes a maximum of 512 bytes.	Specify a shorter template description.
CS.18009	400	The template name includes a maximum of 64 bytes.	Specify a shorter template name.
CS.19001	400	The JSON format of the request body is incorrect.	Refer to the *Cloud Stream Service API Reference* or JSON format specifications.
CS.19002	400	Failed to authorize OBS bucket {Bucket name}.	Submit a service ticket for troubleshooting.

Error Code	Status Code	Description	Solution
CS.19003	400	The bucket to be authorized does not exist.	Use a correct OBS bucket.
CS.19004	400	No OBS bucket name in the request body.	Add the OBS bucket name in JSON array format to the request body.
CS.19005	400	The request contains names of authorized OBS buckets.	Filter out names of authorized OBS buckets.
CS.19006	400	An agency named **cloud_stream_agency** already exists but is not authorized to CS.	Modify the agency configuration or delete the agency on the IAM console.
CS.19007	400	No agency is created for CS.	Create an agency.
CS.19008	400	Failed to create an agency.	Submit a service ticket for troubleshooting.
CS.19009	400	Failed to grant permissions to the agency.	Submit a service ticket for troubleshooting.
CS.19010	400	Insufficient permissions granted to the agency.	Proceed to the authorization.
CS.19011	400	Insufficient agency quota.	Go to the IAM console and delete unused agencies.
CS.20001	400	Failed to create a cluster.	Try again later or submit a service ticket for troubleshooting.
CS.20002	400	The JSON format of the request body is incorrect.	Refer to the *Cloud Stream Service API Reference* or JSON format specifications.
CS.20003	400	A maximum of 400 SPUs can be configured for a cluster.	Reduce the maximum number of SPUs configured for a cluster to 400 or less.
CS.20004	400	The total number of available SPUs for all clusters is {0}.	Reduce the cluster quota to {0} or less.
CS.20005	400	The cluster does not exist.	Specify an existing cluster ID.
CS.20006	400	The cluster name exists.	Select a new cluster name.

Error Code	Status Code	Description	Solution
CS.20007	400	The number {0} of SPUs that have been used by a cluster exceeds the maximum SPU quota.	Specify a value greater than {0} for the maximum SPU quota of the cluster.
CS.20008	400	The cluster name cannot be left blank.	Specify a cluster name.
CS.20009	400	The maximum length of a cluster name is 100 bytes.	Specify a shorter cluster name.
CS.20010	400	The maximum length of a cluster description is 512 bytes.	Specify a shorter cluster description.
CS.20011	400	A cluster name consists of letters, digits, underscores (_), and hyphens (-).	Provide a valid cluster name.
CS.20012	400	The SPU quota must be greater than 0.	Provide an SPU quota greater than 0.
CS.20013	400	The maximum SPU quota is 1,000.	Reduce the SPU quota to 1,000 or less.
CS.20014	400	Insufficient resources.	New cluster resources are being prepared. Try again later.
CS.20015	400	Clusters not in the **Running** status cannot be modified.	Modify cluster information when it is running.
CS.20016	400	Failed to stop the cluster due to arrears.	Check, on CS Backend, whether it has stopped.
CS.20017	400	Failed to restore the frozen cluster.	Manually restore the cluster.
CS.20021	400	Check the cluster billing mode and status.	Only pay-per-use clusters in the **Stopped** or **Stopped due to arrears** status can be restarted.
CS.20022	400	Incorrect cluster network configuration.	Modify the network configuration.
CS.20023	400	The **manager_node_spu** parameter setting is incorrect in the request body.	Use a valid **manager_node_spu** setting in the request body.

Error Code	Status Code	Description	Solution
CS.20024	400	Nodes of the specification corresponding to **manager_node_spu** in the request body have been sold out.	Use a valid **manager_node_spu** setting in the request body.
CS.20025	400	Compute nodes of the corresponding specifications have been sold out.	Submit a service ticket.
CS.20026	400	This operation cannot be performed on clusters that are not in the **Running** status.	Check the cluster.
CS.20027	400	The cluster billed on a yearly/monthly basis cannot be manually deleted.	Cancel subscription.
CS.20028	400	The cluster billed on a yearly/monthly basis cannot be manually stopped.	Cancel subscription.
CS.20029	400	The maximum resource quota of a cluster billed on a yearly/monthly basis cannot be modified.	Apply for quota change.
CS.21001	400	Failed to query cluster information.	Try again later or submit a service ticket for troubleshooting.
CS.21011	400	Failed to delete a cluster.	Submit a service ticket for troubleshooting.
CS.21031	400	Failed to obtain the cluster list due to the incorrect **limit** parameter setting.	Set the **limit** parameter to a value greater than 0.
CS.21091	400	The number of used SPUs {0} exceeds the maximum SPU quota.	Specify a value greater than {0} for the maximum SPU quota.
CS.21092	400	The user ID does not exist.	Provide a valid user ID.
CS.21095	400	Failed to obtain the user quota because the invalid **limit** parameter setting.	Set the **limit** parameter to a value greater than 0.
CS.21999	400	Failed to update the SPU quota.	Check the failure cause.

Error Code	Status Code	Description	Solution
CS.22001	400	Incorrect IP address or domain name format.	Provide a correct IP address and domain name.
CS.22002	400	The IP address or domain name already exists.	Ensure that the domain name is unique and the IP address and domain name cannot be the same as those of the cluster node.
CS.22003	400	The domain name ID does not exist.	Provide a correct domain name ID.
CS.22004	400	Incorrect **hosts** file format.	Ensure that each line in the file is in the "ip hostname" format. If an IP address corresponds to multiple domain names, the line can be in the "ip hostname1 hostname2" format.
CS.22005	400	The **hosts** file is empty.	Provide a correct file.
CS.22006	400	The size of the **hosts** file exceeds the upper limit.	Ensure that the size of the **hosts** file is not greater than 8 MB.
CS.22007	400	The number of IP-domain mappings exceeds 1,000.	Delete unused IP-domain mappings.
CS.22101	400	The VPC ID does not exist.	Provide a correct VPC ID.
CS.22102	400	A conflict occurs during the creation of a VPC peering connection.	Provide valid parameters.
CS.22103	400	The ID of the VPC peering connection does not exist.	Provide a valid VPC peering connection ID.
CS.22104	400	The **destination** parameter in the request body is not set.	Set the **destination** parameter in the request body.
CS.22105	400	The VPC peering connection is not in the **ACTIVE** state.	Check the status of the VPC peering connection. If it is in the **Awaiting Acceptance** state, accept the request and try again.

Error Code	Status Code	Description	Solution
CS.22106	400	A conflict occurs when a route is created due to invalid parameters.	Provide valid parameters.
CS.22107	400	The route ID does not exist.	Provide a correct route ID.
CS.22108	400	The **name** parameter is missing in the request body.	Add the **name** parameter to the request body.
CS.23003	400	Failed to query the job list due to the incorrect **root_job_id** parameter setting.	Set the **root_job_id** parameter to a value greater than 0.
CS.24001	400	A maximum of 50 audit logs can be queried.	Query 50 or less audit logs.
CS.24002	400	The resource type of the audit log is job, template, or cluster.	Use the correct resource type.
CS.24003	400	No tracker is found.	Enable CTS.
CS.24010	400	Failed to query audit logs.	Try again later or submit a service ticket for troubleshooting.
CS.25001	400	Failed to query CS overview statistics.	Try again later.
CS.25003	400	Failed to query historical billing information.	Try again later.
CS.26500	400	An internal error occurred on the service.	Submit a service ticket for troubleshooting or try again later.

7.3 Obtaining a Project ID

Obtaining a Project ID by Calling an API

You can obtain a project ID by calling the API used to **Querying Project Information**.

The API for obtaining a project ID is **GET https://{Endpoint}/v3/projects/**. **{Endpoint}** indicates the endpoint of IAM, which can be obtained from **Regions and Endpoints**.

The following is an example response. The value of **id** under **projects** is the project ID.

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "cn-north-4",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
            },
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

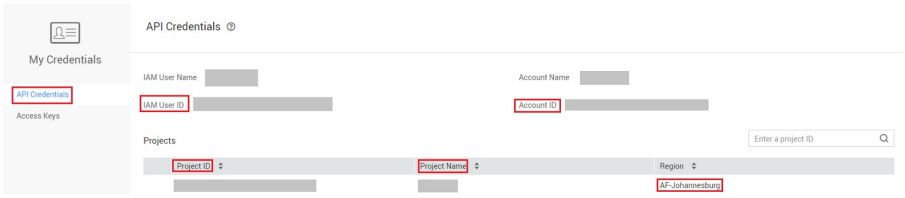
Obtaining a Project ID from the Management Console

A project ID is required for some URIs when an API is called.

To obtain a project ID, perform the following operations:

1. Log in to the management console.
2. Hover the mouse over the username and select **Basic Information**.
3. On the displayed page, click **Manage** in **Security Credentials**.
   On the **Projects** tab of the **API Credentials** page, view project IDs.

   Figure 7-1 Obtaining a Project ID

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

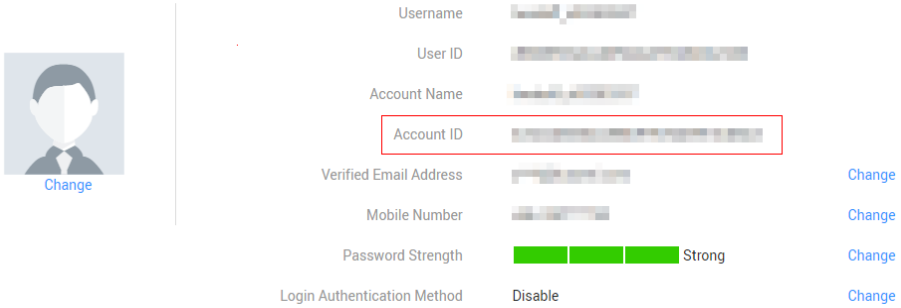
7.4 Obtaining an Account ID

An account ID is required for some URLs when an API is called. To obtain an account ID, perform the following operations:

1. Register an account and log in to the management console. Log in to the management console.
2. Hover the mouse over the username and choose **My Credentials** from the drop-down list.

On the **My Credentials** page, view the **Account ID**. For example, **3c5ea483e62a490b979c9ab8813debb3**.

Figure 7-2 Obtaining an account ID



8 Change History

Released On	Description
2019-05-28	This issue is the twentieth official release. Added the following sections: • **Before You Start** • **Calling APIs** • **Obtaining an Account ID** Modified the following section: **Permissions Policies and Supported Actions**
2019-05-15	This issue is the nineteenth official release. Added the following section: **Querying the Current Release Version of CS** Modified the following sections: • **Creating a SQL Job** • **Updating a SQL Job** • **APIs Related to Job Template Management** • **Error Codes**
2019-04-03	This issue is the eighteenth official release. Added the following sections: • **Running Jobs in Batches** • **Stopping Jobs in Batches** • **Deleting Jobs in Batches**
2019-03-19	This issue is the seventeenth official release. Added the following section: **Permissions Policies and Supported Actions**

Released On	Description
2019-02-26	This issue is the sixteenth official release. Modified the following section: **APIs**
2019-01-10	This issue is the fifteenth official release. Modified the following section: **Authentication**
2018-12-12	This issue is the fourteenth official release. Modified the following sections: • **Querying Job Details**
2018-11-15	This issue is the thirteen official release. Modified the following sections: • **Running a Job** • **Stopping a Job**
2018-10-16	This issue is the twelfth official release. Modified the following sections: • **Before You Start** • **Obtaining a Project ID**
2018-09-25	This issue is the eleventh official release. Modified the following sections: • **API Overview** • **Querying the List of VPC Peering Connections** • **Querying a VPC Peering Connection** • **Deleting a VPC Peering Connection** • **Creating a Route to the Local VPC** • **Query the Route List** • **Deleting a Route**

Released On	Description
2018-08-21	This issue is the tenth official release. Added the following sections: • **Add an IP-Domain Mapping** • **Updating the IP-Domain Mapping** • **Querying an IP-Domain Mapping** • **Deleting an IP-Domain Mapping** • **Terminating an Exclusive Cluster** • **Restarting an Exclusive Cluster** • **Creating a VPC Peering Connection** • **Querying the List of VPC Peering Connections** • **Querying a VPC Peering Connection** • **Deleting a VPC Peering Connection** • **Creating a Route to the Local VPC** • **Query the Route List** • **Deleting a Route** Modified the following sections: • 5.1-APIs Related to Cluster • 5.2-APIs Related to Job Operation • 5.4-APIs Related to Job Query
2018-06-11	This issue is the ninth official release. Modified the following sections: • **Creating a User-Defined Job** • **Updating a User-Defined Job**
2018-05-17	This issue is the eighth official release. The following changes have been made: 5-APIs: Updated the error code information.
2018-05-11	This issue is the seventh official release. The following changes have been made: • Modified the following sections: – **Creating a SQL Job** – **Updating a SQL Job** – **Querying the List of Jobs** – **Querying Job Details**

Released On	Description
2018-03-30	This issue is the sixth official release. The following changes have been made: ● Modified the following section: – 6-Constant
2018-03-07	This issue is the fifth official release. The following changes have been made: ● Modified the following sections: – 5.2-APIs Related to Job Operation – 5.4-APIs Related to Job Query
2018-02-02	This issue is the fourth official release. The following changes have been made: ● Added the following sections: – **Before You Start** – 4-API Usage ● Modified the following section: – 5-APIs
2018-01-12	This issue is the third official release. The following changes have been made: ● Added the following section: – **Error Codes**
2018-01-05	This issue is the second official release. The following changes have been made: ● Modified the following sections: – 5.2-APIs Related to Job Operation – 5.3-APIs Related to Job Template Operation – 5.4-APIs Related to Job Query
2017-12-21	This issue is the first official release.

