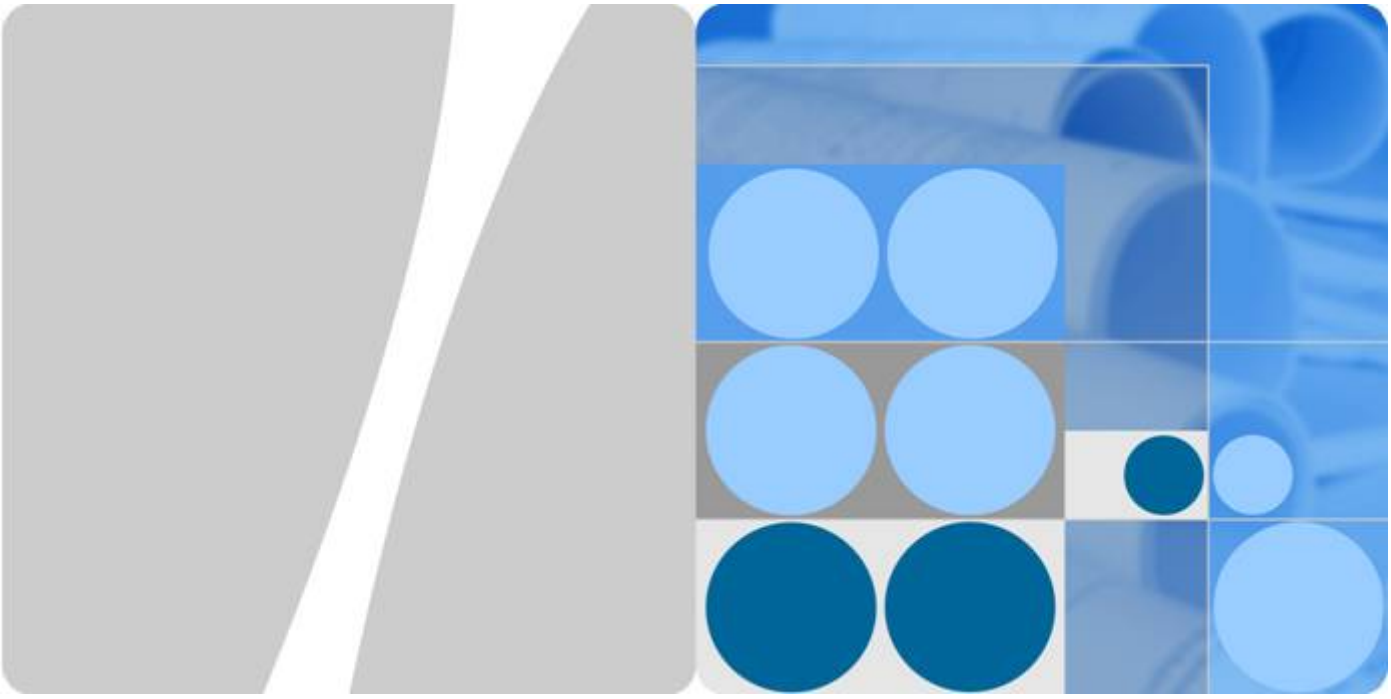


DataLake Governance Center

API Reference

Date **2020-09-02**

Contents

1 Before You Start 1
1.1 Overview 1
1.2 API Calling 2
1.3 Endpoints 2
1.4 Restrictions and Limitations 3
1.5 Concepts 3
2 API Overview 5
3 Calling APIs 6
3.1 Making an API Request 6
3.2 Authentication 11
3.3 Response 12
4 Data Development APIs 14
4.1 Connection Management APIs 14
4.1.1 Creating a Connection 14
4.1.2 Querying a Connection List 24
4.1.3 Viewing Connection Details 26
4.1.4 Editing a Connection 28
4.1.5 Deleting a Connection 31
4.1.6 Exporting a Connection 32
4.1.7 Importing a Connection 33
4.2 Script Development APIs 36
4.2.1 Creating a Script 36
4.2.2 Querying a Script 40
4.2.3 Querying a Script List 43
4.2.4 Modifying a Script 47
4.2.5 Deleting a Script 51
4.2.6 Executing a Script 52
4.2.7 Querying the Execution Result of a Script Instance 54
4.2.8 Stopping Executing a Script Instance 57
4.3 Resource Management APIs 58
4.3.1 Creating a Resource 58
4.3.2 Querying a Resource 61

4.3.3 Querying a Resource List 64
4.3.4 Modifying a Resource 66
4.3.5 Deleting a Resource 69
4.4 Job Development APIs 70
4.4.1 Creating a Job 70
4.4.2 Editing a Job 94
4.4.3 Viewing a Job List 97
4.4.4 Viewing Job Details 100
4.4.5 Exporting a Job 103
4.4.6 Batch Exporting Jobs 105
4.4.7 Viewing a Job File 107
4.4.8 Importing a Job 110
4.4.9 Starting a Job 112
4.4.10 Stopping a Job 114
4.4.11 Deleting a Job 115
4.4.12 Executing a Job Immediately 116
4.4.13 Viewing Running Status of a Real-Time Job 118
4.4.14 Viewing a Job Instance List 122
4.4.15 Viewing Job Instance Details 127
4.4.16 Stopping a Job Instance 130
4.4.17 Retry a Job Instance 131
4.4.18 Querying a System Task 132
4.5 Data Structure 136
5 Appendix 138
5.1 Status Codes 138
5.2 Error Codes 142
5.3 Obtaining a Project ID 143
5.4 Obtaining an Account ID 144
5.5 Obtaining a Job ID 145
5.6 Parsing a Stream in a Response Message 145
A Change History 147

1 Before You Start

1.1 Overview

Background

Enterprises often have to face the following challenges when managing data:

- Data governance
  - There is no methodology available for defining enterprise data system standards and data specifications, and data languages are not unified.
  - There are no efficient and accurate data search tools available to ordinary business personnel, so data is hard to find.
  - There are no associations formed between technical metadata and business metadata, so data cannot be read.
  - There is no quality control or evaluation methods available, so data cannot be trusted.
- Data operations
  - There is no efficient data operations platform to address the growing report analysis requirements and the decision-makers' operational demands.
  - Data is not service-oriented, so new copies are constantly being created and the data definitions are inconsistent. The same data developed repeatedly, wasting resources and driving up costs.
- Data innovation
  - There are too many data silos. Data is not easily shared and circulated throughout enterprises. As a result, cross-domain data analysis and data innovation are lacking.
  - Data is still mainly used in report analysis. There are almost no solutions for promoting business innovation based on data feedback.

What Is Data Lake Governance Center?

DGC is a one-stop data operations platform that helps enterprises manage their data with intelligence and drives enterprise digital operations. This platform

includes components such as Data Integration, Data Design, Data Development, Data Quality Control, Data Assets, and Data Lake Mall. Enterprises can use DGC to build intelligent industrial knowledge bases and end-to-end data systems. Integrating diversified data foundations such as big data computing, analysis, and storage engines, DGC can eliminate data silos, unify data standards, facilitate data monetization, and accelerate digital transformation.

This document describes how to call application programming interfaces (APIs) to perform operations, such as performing or stopping jobs, on Data Lake Governance Center. For details on all supported operations, see **API Overview**.

Before calling a Data Lake Governance Center API, ensure that you have fully understood the concepts related to Data Lake Governance Center. For details, see the ***Service Overview***.

1.2 API Calling

Data Lake Governance Center supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details on API calling, see **Calling APIs**.

1.3 Endpoints

Table 1-1 lists Data Lake Governance Center endpoints. Select one based on the service requirements.

Table 1-1 Endpoint

Region	Endpoint Region	Endpoint	Protocol
CN North-Beijing1	cn-north-1	dgc-dlf.cn-north-1.myhuaweicloud.com	HTTPS
CN North-Beijing4	cn-north-4	dgc-dlf.cn-north-4.myhuaweicloud.com	HTTPS
CN East-Shanghai2	cn-east-2	dgc-dlf.cn-east-2.myhuaweicloud.com	HTTPS
CN East-Shanghai1	cn-east-3	dgc-dlf.cn-east-3.myhuaweicloud.com	HTTPS
CN South-Guangzhou	cn-south-1	dgc-dlf.cn-south-1.myhuaweicloud.com	HTTPS

1.4 Restrictions and Limitations

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

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 users 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.

- User

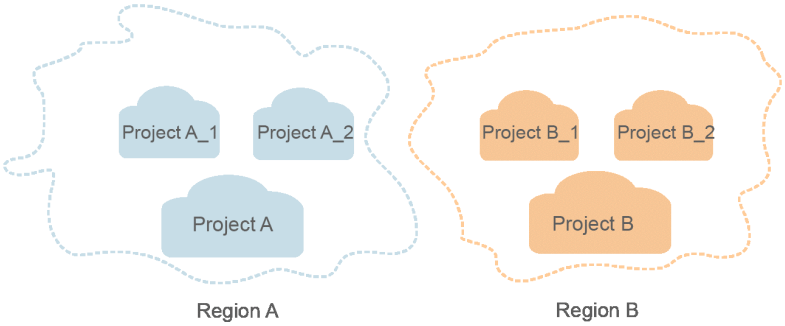
  A user is created using an accounton IAM to use cloud services. Each user has its own identity credentials (password and access keys).

  A user can view the account ID and user ID on the **My Credentials** page of the console. The account, username, and password will be required for API authentication.

- 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. Users can be granted permissions to access all resources in a specific project. If you want more refined access control, create subprojects under a project and create resources in the subprojects. Then, grant users the permissions to access only specific resources in the subprojects.

  Figure 1-1 Project isolating model

- Job

  A job is composed of one or more nodes that are performed collaboratively to complete data operations.

- Node
  A node defines the operations performed on data.

2 API Overview

DGC provides self-developed APIs that comply with RESTful API design specifications. You can call those APIs to perform operations on DGC.

Table 2-1 API description

Type	Description
Connection Management APIs	Used to create, edit, delete, import, and export connections, query the connection list, and query connection details
Script Development APIs	Used to create, query, modify, delete, and execute scripts.
Resource Management APIs	Used to create, query, and modify resources.
Job Development APIs	Used to create, edit, delete, import, and export jobs, query job lists, and query job details.

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.

You can also learn how to construct a request to call an API by watching the video tutorial.

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

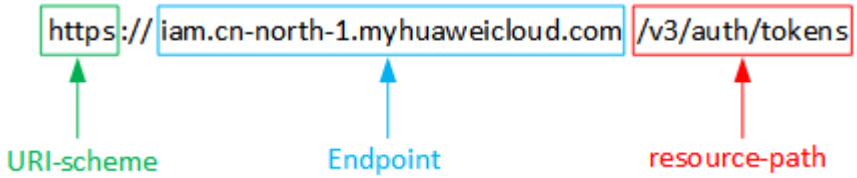
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. **It can be obtained from regions and endpoints**. For example, the endpoint of IAM in the **CN North-Beijing4** region is **iam.cn-north-4.myhuaweicloud.com**.
resource-path	Access path of an API for performing a specified operation. Obtain the value 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 each query parameter that is in the format of "*Parameter name=Parameter value*", for example, **? limit=10**, it indicates that a maximum of 10 data records are allowed.

For example, to obtain an IAM token in the **CN North-Beijing4** region, obtain the endpoint of IAM **(iam.cn-north-4.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-4.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** value of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Method

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-4.myhuaweicloud.com/v3/auth/tokens
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request 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	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of *Hostname:Port number*. If the port number is not specified, the default port is used. The default port number for HTTPS is 443.	No. This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443

Field	Description	Mandatory	Example
Content-Type	Specifies the request body MIME type. This field is mandatory and its default value is **application/json**. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495
X-Project-Id	Specifies a project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies a user token. It is a response to the API used to **obtain a user token**. This API is the only one that does not require authentication. After the request is processed, the value of **X-Subject-Token** in the header is the token value.	No. This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQc-Co...ggg1BBIINPXsidG9rZ
X-Language	Specifies the request language.	Yes	en_us

> **NOTE**
>
> In addition to supporting token-based authentication, the HUAWEI CLOUD APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature authentication) 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-4.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

(Optional) Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

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* (user account), *********** (login password), and *xxxxxxxxxxxxxxxxx* (project ID) with the actual values. To learn how to obtain a project ID, see **Obtaining a Project ID**.

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

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json

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

```
    }
}
```

If all data required for the API request is available, you can send the request to call the 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 authentication is recommended because it is more secure than token 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 certain permissions in a computer system. A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling APIs.

When calling the API to **obtain a user token**, you must set **auth.scope** in the request body to **project**.

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

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

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

You can also learn how to use token-based authentication by watching the video tutorial.

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 on how to sign requests and use the signing SDKs, see ***API Signature Guide***.

NOTICE

The signing SDKs are only used for signing requests and 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 Codes**.

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



(Optional) Response Body

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

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

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "cn-north-4",
...
```

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 Data Development APIs

4.1 Connection Management APIs

4.1.1 Creating a Connection

Function

This API is used to create a connection. The supported connection types include DWS, DLI, SparkSQL, Hive, RDS, and CloudTable.

URI

- URI format

  POST /v1/{project_id}/connections

- Parameter description

  Table 4-1 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**.

Request Message

Table 4-2 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-3 Connection parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the connection. The name consists of a maximum of 100 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The connection name must be unique.
type	Yes	String	Connection type. The options are as follows: • DWS • DLI • SparkSQL • HIVE • RDS • CloudTable
config	No	Map<String,String>	Connection configuration item. The configuration item varies with the connection type. You do not need to set the **config** parameter for the DLI connection. For other types of connections, see the description of connection configuration items.
description	No	String	The description contains a maximum of 255 characters.

Table 4-4 DWS connection parameters

Parameter	Mandatory	Type	Description
clusterName	No	String	DWS cluster name. Perform the following operations to obtain the DWS cluster name: 1. Log in to the management console. 2. Click **Data Warehouse Service** and select **Cluster Management** from the list on the left. You can obtain the cluster name from the cluster management list. By default, this parameter is left blank.
ip	No	String	IP address for accessing the DWS cluster. Perform the following operations to obtain the DWS access address: 1. Log in to the management console. 2. Click **Data Warehouse Service** and select **Cluster Management** from the list on the left. 3. Click the cluster name. The basic information page of the cluster is displayed. You can obtain the private network IP address on the **Database Attribute** tab. If there are multiple IP addresses, select the first IP address. By default, this parameter is left blank.

Parameter	Mandatory	Type	Description
port	No	String	Port for accessing the DWS cluster. Perform the following operations to obtain the DWS access port: 1. Log in to the management console. 2. Click **Data Warehouse Service** and select **Cluster Management** from the list on the left. 3. Click the cluster name. The basic information page of the cluster is displayed. You can obtain the port information on the **Database Attribute** tab. For example, 8000. By default, this parameter is left blank.
userName	Yes	String	Username of the database. The username is required for creating the DWS cluster.
password	Yes	String	Password for accessing the database. The password is required for creating the DWS cluster.
sslEnable	Yes	boolean	Specifies whether to enable the SSL connection.
kmsKey	Yes	String	Name of the KMS key. Perform the following operations to obtain the key: 1. Log in to the management console. 2. Click **Key Management Service** and select **Key Management Service** from the list on the left. You can obtain the key name from the key list.

Parameter	Mandatory	Type	Description
agentName	Yes	String	Name of the CDM cluster. Perform the following operations to obtain the CDM cluster name: 1. Log in to the management console. 2. Click **Cloud Data Migration** and select **Cluster Management** from the list on the left. You can obtain the cluster name from the cluster list.

Table 4-5 Spark SQL connection parameters

Parameter	Mandatory	Type	Description
clusterName	Yes	String	MRS cluster name. Perform the following operations to obtain the MRS cluster name: 1. Log in to the management console. 2. Click **MapReduce Service** and select **Active Clusters** from the list on the left. You can obtain the cluster name from the active clusters, such as **mrsCluster1**.
connectionMethod	Yes	String	Method to connect. • **agent**: connected through an agent. • **direct**: connected directly.
userName	No	String	Username of the MRS cluster. This parameter is mandatory when **connectionMethod** is set to **agent**.
password	No	String	Password for accessing the MRS cluster. This parameter is mandatory when **connectionMethod** is set to **agent**.

Parameter	Mandatory	Type	Description
agentName	No	String	Name of the CDM cluster. This parameter is mandatory when **connectionMethod** is set to **agent**. Perform the following operations to obtain the CDM cluster name: 1. Log in to the management console. 2. Click **Cloud Data Migration** and select **Cluster Management** from the list on the left. You can obtain the cluster name from the cluster list.
kmsKey	No	String	Name of the KMS key. This parameter is mandatory when **connectionMethod** is set to **agent**. Perform the following operations to obtain the key: 1. Log in to the management console. 2. Click **Key Management Service** and select **Key Management Service** from the list on the left. You can obtain the key name from the key list.

Table 4-6 Hive connection parameters

Parameter	Mandatory	Type	Description
clusterName	Yes	String	Name of the MRS cluster. For example, mrsCluster1
connectionMethod	Yes	String	Method to connect. • **agent**: connected through an agent. • **direct**: connected directly.
userName	No	String	Username of the MRS cluster. This parameter is mandatory when **connectionMethod** is set to **agent**.
password	No	String	Password for accessing the MRS cluster. This parameter is mandatory when **connectionMethod** is set to **agent**.

Parameter	Mandatory	Type	Description
agentName	No	String	Name of the CDM cluster. This parameter is mandatory when **connectionMethod** is set to **agent**. Perform the following operations to obtain the CDM cluster name: 1. Log in to the management console. 2. Click **Cloud Data Migration** and select **Cluster Management** from the list on the left. You can obtain the cluster name from the cluster list.
kmsKey	No	String	Name of the KMS key. This parameter is mandatory when **connectionMethod** is set to **agent**. Perform the following operations to obtain the key: 1. Log in to the management console. 2. Click **Key Management Service** and select **Key Management Service** from the list on the left. You can obtain the key name from the key list.

Table 4-7 RDS connection parameters

Parameter	Mandatory	Type	Description
ip	Yes	String	Address for accessing RDS. Perform the following operations to obtain the RDS access address: 1. Log in to the management console. 2. Click **Relational Database Service** and select **Instance Management** from the list on the left. 3. Click the name of an instance. The basic information page of the instance is displayed. You can obtain the IP address on the **Connection Information** tab.
port	Yes	String	Port for accessing RDS. Perform the following operations to obtain the RDS access port: 1. Log in to the management console. 2. Click **Relational Database Service** and select **Instance Management** from the list on the left. 3. Click the name of an instance. The basic information page of the instance is displayed. You can obtain the database port number on the **Connection Information** tab.
userName	Yes	String	Username of the database. The username is required for creating the cluster.
password	Yes	String	Password for accessing the database. The password is required for creating the cluster.

Parameter	Mandatory	Type	Description
kmsKey	Yes	String	Name of the KMS key. Perform the following operations to obtain the key: 1. Log in to the management console. 2. Click **Key Management Service** and select **Key Management Service** from the list on the left. You can obtain the key name from the key list.
agentName	Yes	String	Name of the CDM cluster. Perform the following operations to obtain the CDM cluster name: 1. Log in to the management console. 2. Click **Cloud Data Migration** and select **Cluster Management** from the list on the left. You can obtain the cluster name from the cluster list.
driverName	Yes	String	Name of the driver. • com.mysql.jdbc.Driver • org.postgresql.Driver
driverPath	Yes	String	Path of the driver on OBS.

CloudTable connection parameters

Parameter	Mandatory	Type	Description
clusterName	Yes	String	Name of the CloudTable cluster. Perform the following operations to obtain the cluster name: 1. Log in to the management console. 2. Click **CloudTable Service** and select **Cluster Mode** from the list on the left. You can obtain the cluster name from the cluster list.

Response Message

None.

Example

Create a connection.

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections
{
   "name":"connection1",
   "type":"DWS",
   "config":{
      "clusterName":"test",
      "userName":"dbadmin",
          "password":"test1234",
      "kmsKey":"cdm-dlf",
      "agentName":"cdm-donotdelete",
      "sslEnable":false
   }
}
```

- Success response

  HTTP status code 204

- Failure response

  HTTP status code 400

```
{
   "error_code":"DLF.6309",
   "error_msg":"The name already exists."
}
```

Status Codes

For details about status codes, see **Status Codes**.

4.1.2 Querying a Connection List

Function

This API is used to query a connection list.

URI

- URI format

  GET /v1/{project_id}/connections?offset={offset}&limit={limit}&connectionName={connectionName}

- Parameter description

  Table 4-8 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**.
offset	No	Integer	The value is no less than **0**. The default value is **0**.
limit	No	Integer	The maximum number of records on each page. Value range: 1 to 100 Default value: 10
connectionName	No	String	Name of the connection.

Request Message

Table 4-9 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Parameter	Mandatory	Type	Description
total	Yes	Integer	The total number of connections.
connections	Yes	List<Connection>	Connection list.

Table 4-10 Connection parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the connection. The name consists of a maximum of 100 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The connection name must be unique.
type	Yes	String	Connection type. The options are as follows: • DWS • DLI • SparkSQL • HIVE • RDS • CloudTable
config	No	Map<String,String>	Connection configuration item. The configuration item varies with the connection type. You do not need to set the **config** parameter for the DLI connection. For other types of connections, see the description of connection configuration items.
description	No	String	The description contains a maximum of 255 characters.

Example

Query a connection list.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections
  ```

- Success response

  HTTP status code 200

```
{
   "total":1,
   "connections":[
      {
         "name":"connection1",
         "type":"DWS",
         "config":{
            "clusterName":"test",
            "userName":"dbadmin",
            "password":"test1234",
            "kmsKey":"cdm-dlf",
            "agentName":"cdm-donotdelete",
            "sslEnable":false
         }
      }
   ]
}
```

- Failure response

  HTTP status code 400

```
{
   "error_code":"DLF.3051",
   "error_msg":"The request parameter is invalid."
}
```

4.1.3 Viewing Connection Details

Function

This API is used to query configuration details of a specific connection.

URI

- URI format

  GET /v1/{project_id}/connections/{connection_name}

- Parameter description

  Table 4-11 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**.
connection_name	Yes	String	Name of the connection.

Request Message

Table 4-12 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-13 Connection parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the connection. The name consists of a maximum of 100 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The connection name must be unique.
type	Yes	String	Connection type. The options are as follows: • DWS • DLI • SparkSQL • HIVE • RDS • CloudTable
config	No	Map<String,String>	Connection configuration item. The configuration item varies with the connection type. You do not need to set the **config** parameter for the DLI connection. For other types of connections, see the description of connection configuration items.

Parameter	Mandatory	Type	Description
description	No	String	The description contains a maximum of 255 characters.

Example

Query connection details.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/connection1
  ```
- Success response

  HTTP status code 200

  ```
  {
     "name":"connection1",
     "type":"DWS",
     "config":{
        "clusterName":"test",
        "userName":"dbadmin",
            "password":"test1234",
        "kmsKey":"cdm-dlf",
        "agentName":"cdm-donotdelete",
        "sslEnable":false
     }
  }
  ```
- Failure response

  HTTP status code 400

  ```
  {
     "error_code":"DLF.6322",
     "error_msg":"The data connection does not exist."
  }
  ```

Status Codes

For details about status codes, see **Status Codes**.

4.1.4 Editing a Connection

Function

This API is used to edit a connection.

URI

- URI format

  PUT /v1/{project_id}/connections/{connection_name}?ischeck=true
- Parameter description

Table 4-14 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**.
connection_name	Yes	String	Name of the connection.
ischeck	No	String	Indicates whether to perform check. The default value is **No**.

Request Message

Table 4-15 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-16 Connection parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the connection. It contains a maximum of 100 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The connection name must be unique.

Parameter	Mandatory	Type	Description
type	Yes	String	Connection type. The options are as follows: • DWS • DLI • SparkSQL • HIVE • RDS • CloudTable
config	Yes	Map<String,String>	Connection configuration item. The configuration item varies with the connection type. For details, see the description of the connection configuration items.
description	No	String	The description contains a maximum of 255 characters.

Response Message

None.

Example

Modify a connection.

- Request

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/connection1?ischeck=true
{
   "name":"connection1",
   "type":"DWS",
   "config":{
      "clusterName":"test",
      "userName":"dbadmin",
         "password":"test1234",
      "kmsKey":"cdm-dlf",
      "agentName":"cdm-donotdelete",
      "sslEnable":false
   }
}
```

- Success response

  HTTP status code 204

- Failure response

  HTTP status code 400

```
{
   "error_code":"DLF.6322",
   "error_msg":"The data connection does not exist."
}
```

Status Codes

For details about status codes, see **Status Codes**.

4.1.5 Deleting a Connection

Function

This API is used to delete a connection.

URI

- URI format
  DELETE /v1/{project_id}/connections/{connection_name}
- Parameter description

Table 4-17 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**.
connection_name	Yes	String	Name of the connection.

Request Message

Table 4-18 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

None.

Example

Delete the connection.

- Request
  DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/connection1
- Success response

  HTTP status code 204

4.1.6 Exporting a Connection

Function

This API is used to export all connection information that is compressed in ZIP format.

URI

- URI format

  POST /v1/{project_id}/connections/export
- Parameter description

  Table 4-19 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**.

Request Message

Table 4-20 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

The value of **Content-Type** in the response message is **application/octet-stream** that needs to be parsed and converted into a file. For details, see **Parsing a Stream in a Response Message**. Response messages are compressed as a file. The file name format is **DLF_All_DataConnections.zip**. The file directory is as follows:

```
connections
├─{dliConnection}.conn
├─{dwsConnection}.conn
```

Example

Export a connection.

- Request
  ```
  POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/export
  ```
- Success response

  HTTP status code 200

  ```
  Response messages are compressed as a file. The file directory is as follows:
  connections
  ├─{dliConnection}.conn
  ├─{dwsConnection}.conn
  ```
- Failure response

  HTTP status code 400

  ```
  {
      "error_code":"DLF.6322",
      "error_msg":"The data connection does not exist."
  }
  ```

Status Codes

For details about status codes, see **Status Codes**.

4.1.7 Importing a Connection

Function

This API is used to import one or more connection files from OBS to the Data Development module. Before using this API, store connection files in OBS buckets.

URI

- URI format

  POST /v1/{project_id}/connections/import
- Parameter description

Table 4-21 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**.

Request Message

The parameters are described as follows:

Parameter	Mandatory	Type	Description
path	Yes	String	With OBS deployed: OBS path for storing the connection definition file. For details about the format of the job definition file, see the response message of the exported connection. Without OBS deployed deployed: local path for storing the connection definition file.
params	No	List<Params>	Connection parameter. By default, this parameter is left blank.
sameNamePolicy	No	String	Policy for specifying how to handle duplicate names. The options are as follows: • **SKIP** • **OVERWRITE** The default value is **SKIP**.

Params connection parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the connection.
type	Yes	String	Connection type.

Parameter	Mandatory	Type	Description
params	No	Map<String,Stri ng>	Connection parameter. For details about parameter names, see the description of each type of connection configuration item. By default, this parameter is left blank.

Response Message

Parameter	Mandatory	Type	Description
taskId	Yes	String	ID of the task. Used to call the API for querying system tasks to obtain the import status.

Example

Import a connection.

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/connections/import
{
   "path":"obs://00zyx/2019-07-02/DLF_All_DataConnections.zip",
   "sameNamePolicy":"OVERWRITE",
   "params":[
      {
         "name":"cloudTable",
         "type":"CloudTable",
         "params":{
            "clusterName":"cluster1"
         }
      },
      {
         "name":"hive",
         "type":"HIVE",
         "params":{
            "clusterName":"mrs_ymcc",
            "connectionMethod":"agent",
            "userName":"admin",
            "agentName":"cdm-donotdelete",
            "kmsKey":"KMS-42ab"
         }
      }
   ]
}
```

- Success response

  HTTP status code 200

```
{
   "taskId":"008aae2e675933c7016759418e870000"
}
```

- Failure response

  HTTP status code 400

  ```
  {
     "error_code":"DLF.0815",
     "error_msg":"Fail to read OBS file."
  }
  ```

Status Codes

For details about status codes, see **Status Codes**.

4.2 Script Development APIs

4.2.1 Creating a Script

Function

This API is used to create a script. Currently, the following script types are supported: DLI SQL, Flink SQL, Spark SQL, Hive SQL, DWS SQL, RDS SQL, and Shell.

URI

- URI format

  POST /v1/{project_id}/scripts

- Parameter description

  Table 4-22 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**.

Request Message

Table 4-23 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-24 Script parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the script. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), and periods (.) are allowed. The script name must be unique.
type	Yes	String	Script type. The value can be: • FlinkSQL • DLISQL • SparkSQL • HiveSQL • DWSSQL • RDSSQL • Shell
content	Yes	String	Script content. A maximum of 64 KB is supported.

Parameter	Mandatory	Type	Description
directory	No	String	Directory for storing the script. Perform the following operations to obtain the directory: 1. Log in to the management console. 2. Click **Data Lake Factory** and choose **Development** > **Develop Script** from the list on the left. In the directory tree of the script, you can view the created directories. The default directory is the **root** directory.
connectionName	No	String	Name of the connection associated with the script. This parameter is mandatory when **type** is set to DLI SQL, Spark SQL, Hive SQL, DWS SQL, or Shell. To obtain the existing connections, refer to the instructions in **Querying a Connection List**. By default, this parameter is left blank.
database	No	String	Database associated with an SQL statement. This parameter is available only when **type** is set to DLI SQL, Spark SQL, Hive SQL, or DWS SQL. • If **type** is set to **DLI SQL**, obtain database information by calling the API for **Querying All Databases**. • If **type** is not set to **DLI SQL**, connect to the cluster in JDBC mode to query database information. By default, this parameter is left blank.

Parameter	Mandatory	Type	Description
queueName	No	String	Queue name of the DLI resource. This parameter is available only when **type** is set to **DLI SQL**. You can obtain the queue information by calling the API for **Querying All Queues**. By default, this parameter is left blank.
configuration	No	map <String,Object >	Configuration defined by a user for the job. This parameter is available only when **type** is set to **DLI SQL**. For details about the supported configuration items, see **conf parameter description**. By default, this parameter is left blank.
description	No	String	The description contains a maximum of 255 characters.

Response Message

None.

Example

Create a script.

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts
{
"name":"echoTimeShell",
"type":"Shell",
"content":"echo a",
"connectionName":"con"
}
```

- Success response

  HTTP status code 204

- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.6247",
    "error_msg":"The script type is not specified."
}
```

Status Code

For details about status codes, see **Status Codes**.

4.2.2 Querying a Script

Function

This API is used to query a script, including the script type and script content.

URI

- URI format
  GET /v1/{project_id}/scripts/{script_name}
- Parameter description

Table 4-25 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**.
script_name	Yes	String	Name of the script.

Request Message

Table 4-26 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-27 Script parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the script. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), and periods (.) are allowed. The script name must be unique.
type	Yes	String	Script type. The value can be: • FlinkSQL • DLISQL • SparkSQL • HiveSQL • DWSSQL • RDSSQL • Shell
content	Yes	String	Script content. A maximum of 64 KB is supported.
directory	No	String	Directory for storing the script. Perform the following operations to obtain the directory: 1. Log in to the management console. 2. Click **Data Lake Factory** and choose **Development** > **Develop Script** from the list on the left. In the directory tree of the script, you can view the created directories. The default directory is the **root** directory.

Parameter	Mandatory	Type	Description
connectionName	No	String	Name of the connection associated with the script. This parameter is mandatory when **type** is set to DLI SQL, Spark SQL, Hive SQL, DWS SQL, or Shell. To obtain the existing connections, refer to the instructions in **Querying a Connection List**. By default, this parameter is left blank.
database	No	String	Database associated with an SQL statement. This parameter is available only when **type** is set to DLI SQL, Spark SQL, Hive SQL, or DWS SQL. • If **type** is set to **DLI SQL**, obtain database information by calling the API for **Querying All Databases**. • If **type** is not set to **DLI SQL**, connect to the cluster in JDBC mode to query database information. By default, this parameter is left blank.
queueName	No	String	Queue name of the DLI resource. This parameter is available only when **type** is set to **DLI SQL**. You can obtain the queue information by calling the API for **Querying All Queues**. By default, this parameter is left blank.
configuration	No	map <String,Object>	Configuration defined by a user for the job. This parameter is available only when **type** is set to **DLI SQL**. For details about the supported configuration items, see **conf parameter description**. By default, this parameter is left blank.
description	No	String	The description contains a maximum of 255 characters.

Example

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/scriptName
  ```
- Success response

  HTTP status code 200
  ```
  {
  "name":"scriptName",
  "type":"Shell",
  "content":"echo a",
  "connectionName":"con"
  }
  ```
- Failure response

  HTTP status code 400
  ```
  {
     "error_code":"DLF.6201",
     "error_msg":"The script does not exist."
  }
  ```

Status Code

For details about status codes, see **Status Codes**.

4.2.3 Querying a Script List

Function

This API is used to query the script list. A maximum of 1,000 scripts can be returned for each query.

URI

- URI format

  GET /v1/{project_id}/scripts?
  offset={offset}&limit={limit}&scriptName={scriptName}

- Parameter description

  Table 4-28 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**.
offset	No	Integer	Start page of the paging list. Default value: **0** The value must be greater than or equal to **0**.

Parameter	Mandatory	Type	Description
limit	No	Integer	The maximum number of records on each page. The value ranges from 1 to 100. Default value: 10
scriptName	No	String	Name of the script.

Request Message

Table 4-29 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Parameter	Mandatory	Type	Description
total	Yes	Integer	The total number of scripts.
scripts	Yes	List<Script>	A list of scripts.

Table 4-30 Script parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the script. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), and periods (.) are allowed. The script name must be unique.

Parameter	Mandatory	Type	Description
type	Yes	String	Script type. The value can be: • FlinkSQL • DLISQL • SparkSQL • HiveSQL • DWSSQL • RDSSQL • Shell
content	Yes	String	Script content. A maximum of 64 KB is supported.
directory	No	String	Directory for storing the script. Perform the following operations to obtain the directory: 1. Log in to the management console. 2. Click **Data Lake Factory** and choose **Development** > **Develop Script** from the list on the left. In the directory tree of the script, you can view the created directories. The default directory is the **root** directory.
connectionName	No	String	Name of the connection associated with the script. This parameter is mandatory when **type** is set to DLI SQL, Spark SQL, Hive SQL, DWS SQL, or Shell. To obtain the existing connections, refer to the instructions in **Querying a Connection List**. By default, this parameter is left blank.

Parameter	Mandatory	Type	Description
database	No	String	Database associated with an SQL statement. This parameter is available only when **type** is set to DLI SQL, Spark SQL, Hive SQL, or DWS SQL. • If **type** is set to **DLI SQL**, obtain database information by calling the API for **Querying All Databases**. • If **type** is not set to **DLI SQL**, connect to the cluster in JDBC mode to query database information. By default, this parameter is left blank.
queueName	No	String	Queue name of the DLI resource. This parameter is available only when **type** is set to **DLI SQL**. You can obtain the queue information by calling the API for **Querying All Queues**. By default, this parameter is left blank.
configuration	No	map <String,Object >	Configuration defined by a user for the job. This parameter is available only when **type** is set to **DLI SQL**. For details about the supported configuration items, see **conf parameter description**. By default, this parameter is left blank.
description	No	String	The description contains a maximum of 255 characters.

Example

Query a script list.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts
  ```
- Success response

  HTTP status code 200

```
{
   "total":1,
   "connections":[
      {
         "name":"handleTimeShell",
         "type":"Shell",
         "content":"echo a",
         "connectionName":"con"
      }
   ]
}
```

- Failure response

  HTTP status code 400

```
{
   "error_code":"DLF.3051",
   "error_msg":"The request parameter is invalid."
}
```

4.2.4 Modifying a Script

Function

This API is used to modify the configuration items or script contents of a script.

- When modifying a script, you need to specify the name of the script to be modified.
- The script name and type cannot be modified.

URI

- URI format

  PUT /v1/{project_id}/scripts/{script_name}

- Parameter description

Table 4-31 URI parameter description

Parameter	Mandatory	Type	Description
projectId	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
script_name	Yes	String	Name of the script.

Request Message

Table 4-32 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-33 Script parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the script. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), and periods (.) are allowed. The script name must be unique.
type	Yes	String	Script type. The value can be: • FlinkSQL • DLISQL • SparkSQL • HiveSQL • DWSSQL • RDSSQL • Shell
content	Yes	String	Script content. A maximum of 64 KB is supported.

Parameter	Mandatory	Type	Description
directory	No	String	Directory for storing the script. Perform the following operations to obtain the directory: 1. Log in to the management console. 2. Click **Data Lake Factory** and choose **Development** > **Develop Script** from the list on the left. In the directory tree of the script, you can view the created directories. The default directory is the **root** directory.
connectionName	No	String	Name of the connection associated with the script. This parameter is mandatory when **type** is set to DLI SQL, Spark SQL, Hive SQL, DWS SQL, or Shell. To obtain the existing connections, refer to the instructions in **Querying a Connection List**. By default, this parameter is left blank.
database	No	String	Database associated with an SQL statement. This parameter is available only when **type** is set to DLI SQL, Spark SQL, Hive SQL, or DWS SQL. • If **type** is set to **DLI SQL**, obtain database information by calling the API for **Querying All Databases**. • If **type** is not set to **DLI SQL**, connect to the cluster in JDBC mode to query database information. By default, this parameter is left blank.

Parameter	Mandatory	Type	Description
queueName	No	String	Queue name of the DLI resource. This parameter is available only when **type** is set to **DLI SQL**. You can obtain the queue information by calling the API for **Querying All Queues**. By default, this parameter is left blank.
configuration	No	map <String,Object >	Configuration defined by a user for the job. This parameter is available only when **type** is set to **DLI SQL**. For details about the supported configuration items, see **conf parameter description**. By default, this parameter is left blank.
description	No	String	The description contains a maximum of 255 characters.

Response Message

None.

Example

Modify the script content.

- Request

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/echoTimeShell
{
"content":"echo time",
"connectionName":"con"
}
```

- Success response

  HTTP status code 204

- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.6201",
    "error_msg":"The script does not exist."
}
```

4.2.5 Deleting a Script

Function

This API is used to delete a specific script.

URI

- URI format
  DELETE /v1/{project_id}/scripts/{script_name}
- Parameter description

  Table 4-34 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**.
script_name	Yes	String	Name of the script.

Request Message

Table 4-35 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

None.

Example

- Request
  ```
  DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/echoTime
  ```
- Success response
  HTTP status code 204

- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.6201",
    "error_msg":"The script does not exist."
}
```

Status Code

For details about status codes, see **Status Codes**.

4.2.6 Executing a Script

Function

This API is used to execute specified scripts, including DWS SQL, DLI SQL, Hive SQL, Spark SQL, RDS SQL, Flink SQL, and Presto SQL scripts. A script instance is generated each time the script is executed. You can obtain script execution results using the API for querying script execution results. This API is described in **Querying the Execution Result of a Script Instance**.

URI

- URI format

  POST /v1/{project_id}/scripts/{script_name}/execute

- Parameter description

  Table 4-36 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**.
script_name	Yes	String	Script name.

Request Message

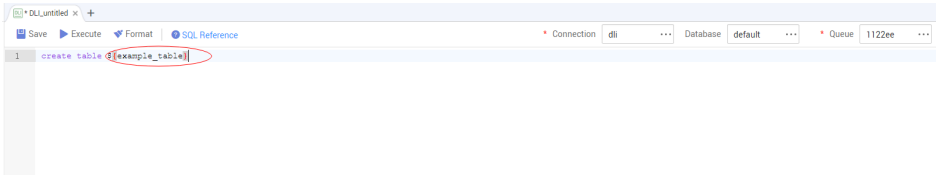
Table 4-37 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-38 Parameter description

Parameter	Mandatory	Type	Description
params	No	Map<String,String>	Script parameter. If a parameter is defined in the script, the parameter value is carried in the params. By default, this parameter is left blank.

Script parameters refer to the parameters in the script content, as shown in the following figure:



Response Message

Table 4-39 Parameter description

Parameter	Mandatory	Type	Description
instanceId	Yes	String	ID of the instance for executing the script. To obtain the execution result, refer to the instructions in **Querying the Execution Result of a Script Instance**.

Example

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/rds_sql/execute
{
"params":{"tableVar":"citys",
"time":"2019-07-25"}
}
```

- Success response

  HTTP status code 200

```
{
"instanceId": "a1ad-448a-9d56-4154193d49c5"
}
```

- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.6201",
    "error_msg":"The script does not exist."
}
```

Status Code

For details about status codes, see **Status Codes**.

4.2.7 Querying the Execution Result of a Script Instance

Function

This API is used to obtain the execution status and result of a script instance.

During the query, you need to specify the script name and script instance ID.

URI

- URI format

  GET /v1/{project_id}/scripts/{script_name}/instances/{instance_id}

- Parameter description

Table 4-40 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**.
script_name	Yes	String	Name of the script.
instance_id	Yes	String	ID of the script instance.

Request Message

Table 4-41 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-42 Parameter description

Parameter	Mandatory	Type	Description
status	Yes	String	Execution status. The options are as follows: **LAUNCHING:** The script instance is being submitted. **RUNNING:** The script instance is running. **FINISHED**: The script instance is successfully run. **FAILED**: The script instance fails to be run.
results	Yes	List<Result>	Execution result of the script instance.

Table 4-43 Result data structure description

Parameter	Mandatory	Type	Description
message	No	String	Execution failure message.
duration	No	float	Duration of executing the script instance. Unit: second
rowCount	No	Int	Number of the result rows.
rows	No	List<List<Object>>	Result data.

Parameter	Mandatory	Type	Description
schema	No	List<Map<String, String>>	Format definition of the result data.

Example

- Request

```
GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/rdsscript/instances/
a1ad-448a-9d56-4154193d49c5
```

- Success response

  HTTP status code 200

```
{
   "results": [{
      "message": "",
            "duration":0.5,
      "rowCount": 1,
      "rows": [[913460.0,
      765.0,
      "8/31/2015 23:26",
      "Harry Bridges Plaza (Ferry Building)",
      50.0,
      "8/31/2015 23:39",
      "San Francisco Caltrain (Townsend at 4th)",
      70.0,
      "288",
      "Subscriber",
      "2139"]],
      "schema": [{
         "TripID": "int"
      },
      {
         "Duration": "int"
      },
      {
         "StartDate": "string"
      },
      {
         "StartStation": "string"
      },
      {
         "StartTerminal": "int"
      },
      {
         "EndDate": "string"
      },
      {
         "EndStation": "string"
      },
      {
         "EndTerminal": "int"
      },
      {
         "Bike": "string"
      },
      {
         "SubscriberType": "string"
      },
      {
         "ZipCode": "string"
      }]
   }],
```

```
    "status": "FINISHED"
}
```

- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.6201",
    "error_msg":"The script does not exist."
}
```

Status Code

For details about status codes, see **Status Codes**.

4.2.8 Stopping Executing a Script Instance

Function

This API is used to stop executing a script instance.

URI

- URI format

  POST /v1/{project_id}/scripts/{script_name}/instances/{instance_id}/stop

- Parameter description

  Table 4-44 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**.
script_name	Yes	String	Name of the script.
instance_id	Yes	String	ID of the script instance.

Request Message

Table 4-45 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

None.

Example

- Request
  ```
  POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/scripts/rdsscript/instances/
  a1ad-448a-9d56-4154193d49c5/stop
  ```
- Success response

  HTTP status code 204
- Failure response

  HTTP status code 400

  ```
  {
      "error_code":"DLF.6205",
      "error_msg":"The script running history does not exist."
  }
  ```

Status Code

For details about status codes, see **Status Codes**.

4.3 Resource Management APIs

4.3.1 Creating a Resource

Function

This API is used to create a resource. Types of nodes, including DLI Spark, MRS Spark, and MRS MapReduce, can reference files such as JAR and properties through resources.

URI

- URI format

  POST /v1/{project_id}/resources
- Parameter description

Table 4-46 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**.

Request Message

Table 4-47 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-48 Resource parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the resource. The name contains a maximum of 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
type	Yes	String	Resource type. The value can be: • archive • file • jar

Parameter	Mandatory	Type	Description
location	Yes	String	OBS path for storing the resource file. When **type** is set to **jar**, **location** is the path for storing the main JAR package. The path contains a maximum of 256 characters. For example, obs://myBucket/test.jar
dependFiles	No	List<String>	JAR package and properties file that the main JAR package depends on. The description contains a maximum of 10,240 characters.
desc	No	String	The description contains a maximum of 255 characters.
directory	No	String	Directory for storing the resource. Perform the following operations to obtain the directory: 1. Log in to the management console. 2. Click **Data Lake Factory** and choose **Configuration** > **Manage Resource** from the list on the left. In the resource directory tree, you can view the created directories. The default value is the **root** directory.

Response Message

Table 4-49 Parameter description

Parameter	Mandatory	Type	Description
resourceId	Yes	String	Resource ID.

Example

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources
{
    "name": "test",
    "type": "jar",
    "location": "obs://00000000dlf-test/hadoop-mapreduce-examples-2.4.1.jar",
    "dependFiles": ["obs://00000000dlf-test/depend1.jar","obs://00000000dlf-test/depend2.jar"],
    "desc": "test",
    "directory":"/resource"
}
```

- Success response

```
{
    "resourceId":"3624d1c3-5df5-4f20-9af9-98eadad6c5f9"
}
```

- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.6259",
    "error_msg":"Files of the same name exist in the directory."
}
```

Status Code

For details about status codes, see **Status Codes**.

4.3.2 Querying a Resource

Function

This API is used to query resource details. A resource contains various files such as JAR, ZIP, and properties files. A created resource can be used in job nodes such as DLI Spark and MRS Spark nodes.

URI

- URI format

  GET /v1/{project_id}/resources/{resource_id}

- Parameter description

Table 4-50 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**.
resource_id	Yes	String	Resource ID.

Request Message

Table 4-51 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-52 Resource parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the resource. The name contains a maximum of 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
type	Yes	String	Resource type. The value can be: • archive • file • jar
location	Yes	String	OBS path for storing the resource file. When **type** is set to **jar**, **location** is the path for storing the main JAR package. The path contains a maximum of 256 characters. For example, obs://myBucket/test.jar
dependFiles	No	List<String>	JAR package and properties file that the main JAR package depends on. The description contains a maximum of 10,240 characters.

Parameter	Mandatory	Type	Description
desc	No	String	The description contains a maximum of 255 characters.
directory	No	String	Directory for storing the resource. Perform the following operations to obtain the directory: 1. Log in to the management console. 2. Click **Data Lake Factory** and choose **Configuration** > **Manage Resource** from the list on the left. In the resource directory tree, you can view the created directories. The default value is the **root** directory.

Example

View resource details.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources/3624d1c3-5df5-4f20-9af9-98eadad6c5f9
  ```
- Success response
  ```
  {
    "name": "test",
    "type": "jar",
    "location": "obs://00000000dlf-test/hadoop-mapreduce-examples-2.4.1.jar",
    "dependFiles": ["obs://00000000dlf-test/depend1.jar","obs://00000000dlf-test/depend2.jar"],
    "desc": "test",
    "directory":"/resource"
  }
  ```
- Failure response

  HTTP status code 400

  ```
  {
    "error_code":"DLF.6241",
    "error_msg":"The resource information does not exist."
  }
  ```

Status Code

For details about status codes, see **Status Codes**.

4.3.3 Querying a Resource List

Function

This API is used to query a resource list. During the query, you can specify the page number and the maximum number of records on each page.

URI

- URI format

  GET /v1/{project_id}/resources?offset={offset}&limit={limit}&resourceName={resourceName}

- Parameter description

  Table 4-53 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**.
offset	No	Integer	Start page of the paging list. Default value: **0** The value must be greater than or equal to **0**.
limit	No	Integer	The maximum number of records on each page. The value ranges from 1 to 100. Default value: 10
resourceName	No	String	Name of the resource.

Request Message

Table 4-54 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Parameter	Mandatory	Type	Description
total	Yes	Integer	The total number of resources.
resources	Yes	List<Resource>	A list of resources.

Table 4-55 Resource parameter description

Parameter	Mandatory	Type	Description
id	Yes	String	ID of the resource. The resource ID is used to query the resource.
name	Yes	String	Name of the resource. The name contains a maximum of 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
type	Yes	String	Resource type. The value can be: ● archive ● file ● jar
location	Yes	String	OBS path for storing the resource file. When **type** is set to **jar**, **location** is the path for storing the main JAR package. The path contains a maximum of 256 characters. For example, obs://myBucket/test.jar
dependFiles	No	List<String>	JAR package and properties file that the main JAR package depends on. The description contains a maximum of 10,240 characters.
desc	No	String	The description contains a maximum of 255 characters.

Parameter	Mandatory	Type	Description
directory	No	String	Directory for storing the resource. Perform the following operations to obtain the directory: 1. Log in to the management console. 2. Click **Data Lake Factory** and choose **Configuration** > **Manage Resource** from the list on the left. In the resource directory tree, you can view the created directories. The default value is the **root** directory.

Example

Query a list of resources.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources
  ```
- Success response

  HTTP status code 200

  ```
  {
     "total":1,
     "resources":[
        {
           "id":"b384b9e9ab9b4ee8994c8633aabc9505"
           "name":"mrResource",
           "type":"jar",
           "location":"obs://00000000dlf-test/hadoop-mapreduce-examples-2.4.1.jar",
           "dependFiles":[
              "obs://00000000dlf-test/depend1.jar",
              "obs://00000000dlf-test/depend2.jar"
           ],
           "desc":"test",
           "directory":"/resource"
        }
     ]
  }
  ```
- Failure response

  HTTP status code 400

  ```
  {
     "error_code":"DLF.3051",
     "error_msg":"The request parameter is invalid."
  }
  ```

4.3.4 Modifying a Resource

Function

This API is used to modify a specific resource. When modifying the resource, specify the resource ID.

- The resource type and directory cannot be modified.

URI

- URI format
  PUT /v1/{project_id}/resources/{resource_id}
- Parameter description

Table 4-56 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**.
resource_id	Yes	String	Resource ID.

Request Message

Table 4-57 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-58 Resource parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the resource. The name contains a maximum of 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
type	Yes	String	Resource type. The value can be: ● archive ● file ● jar
location	No	String	OBS path for storing the resource file. When **type** is set to **jar**, **location** is the path for storing the main JAR package. The path contains a maximum of 256 characters. For example, obs://myBucket/test.jar
dependFiles	No	List<String>	JAR package and properties file that the main JAR package depends on. The description contains a maximum of 10,240 characters.
desc	No	String	The description contains a maximum of 255 characters.
directory	No	String	Directory for storing the resource. Perform the following operations to obtain the directory: 1. Log in to the management console. 2. Click **Data Lake Factory** and choose **Configuration** > **Manage Resource** from the list on the left. In the resource directory tree, you can view the created directories. The default value is the **root** directory.

Response Message

None.

Example

- Request
  ```
  PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources/3624d1c3-5df5-4f20-9af9-98eadad6c5f9
  ```
- Success response

  HTTP status code 204
- Failure response

  HTTP status code 400

  ```
  {
     "error_code":"DLF.6241",
     "error_msg":"The resource information does not exist."
  }
  ```

4.3.5 Deleting a Resource

Function

This API is used to delete a resource.

URI

- URI format
  DELETE /v1/{project_id}/resources/{resource_id}
- Parameter description

  Table 4-59 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**.
resource_id	Yes	String	Resource ID.

Request Message

Table 4-60 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

None.

Example

- Request
  DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/resources/3624d1c3-5df5-4f20-9af9-98eadad6c5f9
- Success response
  HTTP status code 204

- Failure response

  HTTP status code 400

  ```
  {
      "error_code":"DLF.6241",
      "error_msg":"The resource information does not exist."
  }
  ```

Status Code

For details about status codes, see **Status Codes**.

4.4 Job Development APIs

4.4.1 Creating a Job

Function

This API is used to create a job. A job consists of one or more nodes, such as Hive SQL, CS Job, and CDM Job nodes. DLF supports two types of jobs: batch jobs and real-time jobs.

URI

- URI format

  POST /v1/{project_id}/jobs

- Parameter description

  Table 4-61 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**.

Request Message

Table 4-62 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-63 Parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the job. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The job name must be unique.
nodes	Yes	List<Node>	Node definition. For details, see **Table 4-64**.
schedule	Yes	Schedule data structure	Scheduling configuration. For details, see **Table 4-65**.
params	No	List<Param>	Job parameter definition. For details, see **Table 4-66**.
directory	No	String	Directory for saving the job. The value must be an existing directory, for example, /dir/a/. The default value is the **root** directory.
jobType	Yes	String	Job type. The options are as follows: • **REAL_TIME**: real-time processing • **BATCH:** batch processing
basicConfig	No	BasicConfig data structure	Basic job information. For details, see **Table 4-88**.

Table 4-64 Node data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the node. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. Names of the nodes in a job must be unique.

Parameter	Mandatory	Type	Description
type	Yes	String	Node type. The options are as follows: • **Hive SQL**: Runs Hive SQL scripts. • **Spark SQL**: Runs Spark SQL scripts. • **DWS SQL**: Runs DWS SQL scripts. • **DLI SQL**: Runs DLI SQL scripts. • **Shell**: Runs shell SQL scripts. • **CDM Job**: Runs CDM jobs. • **DIS Transfer Task**: Creates DIS dump tasks. • **CS Job:** Creates and starts CloudStream jobs. • **CloudTable Manager**: Manages CloudTable tables, including creating and deleting tables. • **OBS Manager**: Manages OBS paths, including creating and deleting paths. • **REST API**: Sends REST API requests. • **SMN**: Sends short messages or emails. • **MRS Spark**: Runs Spark jobs of MRS. • **MapReduce**: Runs MapReduce jobs of MRS. • **DLI Spark**: Runs Spark jobs of DLF. • **RDS SQL**: Transfers SQL statements to RDS for execution.
location	Yes	Location data structure	Location of a node on the job canvas. For details, see **Table 4-67**.
preNodeName	No	List<String>	Name of the previous node on which the current node depends.

Parameter	Mandatory	Type	Description
conditions	No	List<Condition>	Node execution condition. Whether the node is executed or not depends on the calculation result of the EL expression saved in the expression field of condition. For details, see **Table 4-68**.
properties	Yes	List data structure	Node property. Each type of node has its own property definition, as shown below: • Hive SQL: For details, see **Table 4-73**. • Spark SQL: For details, see **Table 4-74**. • DWS SQL: For details, see **Table 4-75**. • DLI SQL: For details, see **Table 4-76**. • Shell: For details, see **Table 4-77**. • CDM Job: For details, see **Table 4-78**. • DIS TransferTask: For details, see **Table 4-79**. • CS Job: For details, see **Table 4-80**. • CloudTableManager: For details, see **Table 4-81**. • OBSManager: For details, see **Table 4-82**. • RESTAPI: For details, see **Table 4-83**. • SMN: For details, see **Table 4-84**. • MRS Spark: For details, see **Table 4-85**. • MapReduce: For details, see **Table 4-86**. • DLI Spark: For details, see **Table 4-87**.

Parameter	Mandatory	Type	Description
pollingInterval	No	Int	Interval at which node running results are checked. Unit: second; value range: 1 to 60 Default value: 10
maxExecutionTime	No	Int	Maximum execution time of a node. If a node is not executed within the maximum execution time, the node is set to the failed state. Unit: minute; value range: 5 to 1,440 Default value: 60
retryTimes	No	Int	Number of the node retries. The value ranges from 0 to 5. 0 indicates no retry. Default value: 0
retryInterval	No	Int	Interval at which a retry is performed upon a failure. The value ranges from 5 to 120. Unit: second Default value: 120
failPolicy	No	String	Job failure policy. • **FAIL**: Stop the job and set the job to the failed state. • **IGNORE**: Go to the next node. • **SUSPEND**: Suspend the job. The default value is **FAIL**.
eventTrigger	No	Event data structure	Node event triggering configuration. For details, see **Table 4-70**.
cronTrigger	No	Cron data structure	Node Cron triggering configuration. For details, see **Table 4-69**.

Table 4-65 Schedule data structure description

Parameter	Mandatory	Type	Description
type	Yes	String	Scheduling type. • **EXECUTE_ONCE**: The job runs immediately and runs only once. • **CRON**: The job runs periodically. • **EVENT**: The job runs when specified events occur. Currently, only the DIS data triggering event is supported.
cron	No	Data structure	When **type** is set to **CRON**, configure the scheduling frequency and start time. For details, see **Table 4-69**.
event	No	Data structure	When **type** is set to **EVENT**, configure information such as the event source. For details, see **Table 4-70**.

Table 4-66 Param data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Parameter name. It contains a maximum of 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
value	Yes	String	The parameter value cannot be greater than 1,024 characters.
type	No	String	Parameter type. • **variable** • **constants** Default value: **variable**

Table 4-67 Location data structure description

Parameter	Mandatory	Type	Description
x	Yes	Int	Position of the node on the horizontal axis of the job canvas.
y	Yes	Int	Position of the node on the vertical axis of the job canvas.

Table 4-68 condition data structure description

Parameter	Mandatory	Type	Description
preNodeName	Yes	String	Name of the previous node on which the current node depends.
expression	Yes	String	EL expression. If the calculation result of the EL expression is true, this node is executed.

Table 4-69 Cron data structure description

Parameter	Mandatory	Type	Description
startTime	Yes	String	Scheduling start time in the format of yyyy-MM-dd'T'HH:mm:ssZ, which is an ISO 8601 time format. For example, 2018-10-22T23:59:59+08, which indicates that a job starts to be scheduled at 23:59:59 on October 22nd, 2018 in the Beijing time zone.

Parameter	Mandatory	Type	Description
endTime	No	String	Scheduling end time in the format of yyyy-MM-dd'T'HH:mm:ssZ, which is an ISO 8601 time format. For example, 2018-10-22T23:59:59+08, which indicates that a job stops to be scheduled at 23:59:59 on October 22nd, 2018 in the Beijing time zone. If the end time is not set, the job will be executed based on the scheduling period.
expression	Yes	String	Cron expression in the format of <second><minute><hour><day><month><week>. For details about the values input in each field, see Table 7.
expressionTimeZone	No	String	Time zone corresponding to the Cron expression, for example, GMT+8. Default value: Use the time zone where DGC is located.
dependPrePeriod	No	Boolean	Indicates whether to depend on the execution result of the current job's dependent job in the previous scheduling period. Default value: false
dependJobs	No	DependJobs data structure	Job dependency configuration. For details, see **Table 4-72**.

Table 4-70 Event data structure description

Parameter	Mandatory	Type	Description
eventType	Yes	String	Event type. Currently, only newly reported data events from the DIS stream can be monitored. Each time a data record is reported, the job runs once. This parameter is set to **DIS**.

Parameter	Mandatory	Type	Description
channel	Yes	String	DIS stream name. Perform the following operations to obtain the stream name: 1. Log in to the management console. 2. Click **Data Ingestion Service** and choose **Ingestion Management** > **Streams** from the list on the left. 3. The stream management page lists the existing streams.
failPolicy	No	String	Job failure policy. • **SUSPEND**: Suspend the event. • **IGNORE**: Ignore the failure and process with the next event. Default value: **SUSPEND**
concurrent	No	int	Number of the concurrently scheduled jobs. Value range: 1 to 128 Default value: **1**
readPolicy	No	String	Access policy. • **LAST**: Access data from the last location. • **NEW**: Access data from a new location. Default value: LAST

Table 4-71 Values in the Cron expression fields

Field	Value Range	Allowed Special Character	Description
Second	0-59	, - * /	In the current version, only **0** is allowed.
Minute	0-59	, - * /	-
Hour	0-23	, - * /	-

Field	Value Range	Allowed Special Character	Description
Day	1-31	, - * ? / L W C	-
Month	1-12	, - * /	In the current version, only * is allowed.
Week	1-7	, - * ? / L C #	Starting from Sunday.

Table 4-72 DependJobs data structure description

Parameter	Mandatory	Type	Description
jobs	Yes	List<String>	A list of dependent jobs. Only the existing jobs can be depended on.
dependPeriod	No	String	Dependency period. • **SAME_PERIOD**: To run a job or not depends on the execution result of its depended job in the current scheduling period. • **PRE_PERIOD**: To run a job or not depends on the execution result of its depended job in the previous scheduling period. Default value: SAME_PERIOD
dependFailPolicy	No	String	Dependency job failure policy. • **FAIL**: Stop the job and set the job to the failed state. • **IGNORE**: Continue to run the job. • **SUSPEND**: Suspend the job. Default value: **FAIL**

Table 4-73 Parameter description of the Hive SQL node

Parameter	Mandatory	Type	Description
scriptName	Yes	String	Script name.

Parameter	Mandatory	Type	Description
database	No	String	Database name. Database in the MRS Hive. The default value is **default**.
connectionName	No	String	Name of the connection.
scriptArgs	No	String	Script parameter in format of key and value. Multiple parameters are separated by slashes (\), for example, key1=value1\nkey2=value2

Table 4-74 Parameter description of the Spark SQL node

Parameter	Mandatory	Type	Description
scriptName	Yes	String	Script name.
database	No	String	Database name. Database in the MRS Spark SQL. The default value is **default**.
connectionName	No	String	Name of the connection.
scriptArgs	No	String	Script parameter in format of key and value. Multiple parameters are separated by slashes (\), for example, key1=value1\nkey2=value2

Table 4-75 Parameter description of the DWS SQL node

Parameter	Mandatory	Type	Description
scriptName	Yes	String	Script name.
database	No	String	Database name. Database in DWS. The default value is **postgres**.
connectionName	No	String	Name of the connection.

Parameter	Mandatory	Type	Description
scriptArgs	No	String	Script parameter in format of key and value. Multiple parameters are separated by slashes (\), for example, key1=value1\nkey2=value2

Table 4-76 Parameter description of the DLI SQL node

Parameter	Mandatory	Type	Description
scriptName	Yes	String	Script name.
database	No	String	Database name. Database in DLI.
connectionName	No	String	Name of the connection.
scriptArgs	No	String	Script parameter in format of key and value. Multiple parameters are separated by slashes (\), for example, key1=value1\nkey2=value2

Table 4-77 Parameter description of the shell node

Parameter	Mandatory	Type	Description
scriptName	Yes	String	Script name.
connectionName	Yes	String	Name of the connection.
arguments	No	String	Shell script parameter.

Table 4-78 Parameter description of the CDM Job node

Parameter	Mandatory	Type	Description
clusterName	Yes	String	Name of the cluster. Perform the following operations to obtain the CDM cluster name: 1. Log in to the management console. 2. Click **Cloud Data Migration** and select **Cluster Management** from the list on the left. You can obtain the cluster name from the cluster management list.
jobName	Yes	String	Name of the job. Perform the following operations to obtain the job name: 1. Log in to the management console. 2. Click **Cloud Data Migration** and select **Cluster Management** from the list on the left. 3. Click **Job Management** of the cluster to obtain the job name.

Table 4-79 Parameter description of the DISTransferTask node

Parameter	Mandatory	Type	Description
streamName	Yes	String	DIS stream name. Perform the following operations to obtain the stream name: 1. Log in to the management console. 2. Click **Data Ingestion Service** and choose **Ingestion Management** > **Streams** from the list on the left. 3. The stream management page lists the existing streams.
destinationType	Yes	String	Dump target. • CloudTable • OBS
duplicatePolicy	Yes	String	Duplicate name policy. • **OVERWRITE** • **IGNORE**
configuration	Yes	Data structure	Dump configuration. For details, see the description of the **obs_destination_descriptor** and **cloudtable_destination_descriptor** parameters in **Adding a Dump Task**.

Table 4-80 Parameter description of the CS Job node

Parameter	Mandatory	Type	Description
jobName	Yes	String	Name of the CS job. Perform the following operations to obtain the job name: 1. Log in to the management console. 2. Click **Cloud Stream Service** and select **Job Management** from the list on the left. You can obtain the job name from the job management list.
jobType	Yes	String	Job type. ● exists: existing job ● flink_sql_job: Flink SQL job ● flink_jar_job: user-defined Flink job ● spark_streaming_jar_job: user-defined Spark job
jarPath	No	String	JAR package path.
mainClass	No	String	Main class of the JAR package.
jarArgs	No	String	Parameter of the JAR package.
clusterName	No	String	CS cluster name. Perform the following operations to obtain the cluster name: 1. Log in to the management console. 2. Click **Cloud Stream Service** and select **Cluster Management** from the list on the left. You can obtain the cluster name from the cluster management list.

Parameter	Mandatory	Type	Description
spuNumber	No	int	Number of SPUs. 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.
managerSpu	No	int	Number of management units. 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.
parallelNumber	No	int	Number of concurrent running jobs. 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.
executorNumber	No	int	Number of executors. 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.
executorSpu	No	int	Executor SPU. 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.

Parameter	Mandatory	Type	Description
restartWhenException	No	boolean	Specifies whether the CS job automatically restarts when it is abnormal.
udfJar	No	String	UDF JAR path.
scriptName	No	String	Script name.
scriptArgs	No	String	Script parameter in format of key and value. Multiple parameters are separated by slashes (\), for example, key1=value1\nkey2=value2

Table 4-81 Parameter description of the CloudTableManager node

Parameter	Mandatory	Type	Description
namespace	No	String	Namespace. Default value: default
action	Yes	String	Action type. • CREATE_TABLE: Create a table. • DELETE_TABLE: Delete a table.
table	No	String	Table name.
columnFamily	No	String	Column family.

Table 4-82 Parameter description of the OBSManager node

Parameter	Mandatory	Type	Description
action	Yes	String	Action type. • CREATE_PATH: Create an OBS path. • DELETE_PATH: Delete an OBS path.
path	Yes	String	OBS path.

Table 4-83 Parameter description of the RESTAPI node

Parameter	Mandatory	Type	Description
url	Yes	String	URL address. URL of the cloud service.
method	Yes	String	HTTP method. • GET • POST • PUT • DELETE
headers	No	String	HTTP message header in the format of <message header name>=<value>. Multiple message headers are separated by newlines.
body	No	String	Message body.

Table 4-84 Parameter description of the SMN node

Parameter	Mandatory	Type	Description
topic	Yes	String	SMN topic. Perform the following operations to obtain an SMN topic: 1. Log in to the management console. 2. Click **Simple Message Notification** and choose **Topic Management** > **Topics** from the list on the left. You can obtain the SMN topic in the topic list.
subject	Yes	String	Message title, which is used as the subject of an email sent to a subscriber.
messageType	Yes	String	Message type. • NORMAL • STRUCTURE • TEMPLATE
message	Yes	String	Message to be sent.

Table 4-85 Parameter description of the MRS Spark node

Parameter	Mandatory	Type	Description
clusterName	Yes	String	MRS cluster name. Perform the following operations to obtain the MRS cluster name: 1. Log in to the management console. 2. Click **MapReduce Service** and choose **Clusters** > **Active Clusters** from the list on the left. You can obtain the cluster name from the active clusters.
jobName	Yes	String	MRS job name. The job name is user-defined.
resourcePath	Yes	String	Resource path.
parameters	Yes	String	Job parameter.
input	Yes	String	Input path. Input data path of the MapReduce job. The path can be an HDFS or OBS path.
output	Yes	String	Output path. Output data path of the MapReduce job. The path can be an HDFS or OBS path.

Table 4-86 Parameter description of the MapReduce node

Parameter	Mandatory	Type	Description
clusterName	Yes	String	MRS cluster name. Perform the following operations to obtain the MRS cluster name: 1. Log in to the management console. 2. Click **MapReduce Service** and choose **Clusters** > **Active Clusters** from the list on the left. You can obtain the cluster name from the active clusters.

Parameter	Mandatory	Type	Description
jobName	Yes	String	MRS job name. The job name is user-defined.
resourcePath	Yes	String	Resource path.
parameters	Yes	String	Job parameter.
input	Yes	String	Input path. Input data path of the MapReduce job. The path can be an HDFS or OBS path.
output	Yes	String	Output path. Output data path of the MapReduce job. The path can be an HDFS or OBS path.

Table 4-87 Parameter description of the DLI Spark node

Parameter	Mandatory	Type	Description
clusterName	Yes	String	Name of the DLI cluster. Perform the following operations to obtain the DLI cluster name: 1. Log in to the management console. 2. Click **Data Lake Insight** and then **Spark Jobs**. 3. Choose **Cluster Management**. You can obtain the cluster name from the cluster management list.

Parameter	Mandatory	Type	Description
jobName	Yes	String	DLI job name. Perform the following operations to obtain the DLI cluster name: 1. Log in to the management console. 2. Click **Data Lake Insight** and then **Spark Jobs**. 3. Choose **Job Management**. You can obtain the job name from the job management list.
resourceType	No	String	Resource type of the DLI job. **CUSTOMIZED** is returned when the parameter is customized.
jobClass	Yes	String	Main class of the job.
resourcePath	Yes	String	JAR package resource path.
jarArgs	No	String	Main-class entry parameter.
sparkConfig	No	String	Running parameter of the Spark job.

Table 4-88 BasicConfig job information

Parameter	Mandatory	Type	Description
owner	No	String	Job owner. The length cannot exceed 128 characters.
priority	No	int	Job priority. The value ranges from 0 to 2. The default value is **0**. **0** indicates a top priority, **1** indicates a medium priority, and **2** indicates a low priority.
executeUser	No	String	Job execution user. The value must be an existing HUAWEI CLOUD username.
instanceTimeout	No	int	Instance timeout interval. The unit is minute. The value ranges from 5 to 1,440. The default value is **60**.

Parameter	Mandatory	Type	Description
customFields	No	Map<String,String>	User-defined field. The length cannot exceed 2,048 characters.

Response Message

None.

Example

Create a real-time job. The job includes one DLI node (**TRIP_RAW_STANDARD**), two DIS nodes (**DIS_INPUT** and **DIS_EVENT**), and one CS Job (**CS_PROCESS**). The following figure shows the details:

Figure 4-1 Real-time job



- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs
{
    "logPath":"obs://dlf-log-b384b9e9ab9b4ee8994c8633aabc9505",
    "name":"myJob",
    "nodes":[
        {
            "failPolicy":"FAIL",
            "location":{
                "x":"385.0",
                "y":"150.0"
            },
            "maxExecutionTime":60,
            "name":"DIS_INPUT",
            "pollingInterval":1,
            "preNodeName":[],
            "properties":[
                {
                    "name":"streamName",
                    "value":"csinput"
                }
            ],
            "resources":[],
            "retryInterval":120,
            "retryTimes":0,
            "type":"DISStream"
        },
        {
            "failPolicy":"FAIL",
            "location":{
                "x":"572.0",
                "y":"151.0"
```

```
    },
    "maxExecutionTime":60,
    "name":"CS_PROCESS",
    "pollingInterval":10,
    "preNodeName":[
       "DIS_INPUT"
    ],
    "properties":[
       {
          "name":"scriptName",
          "value":"CS_PROCESS_TRIP"
       },
       {
          "name":"jobName",
          "value":"CS_PROCESS"
       },
       {
          "name":"jobType",
          "value":"flink_sql_job"
       },
       {
          "name":"spuNumber",
          "value":"2"
       },
       {
          "name":"parallelNumber",
          "value":"1"
       }
    ],
    "resources":[],
    "retryInterval":120,
    "retryTimes":0,
    "type":"CSJob"
 },
 {
    "failPolicy":"FAIL",
    "location":{
       "x":"718.0",
       "y":"121.0"
    },
    "maxExecutionTime":60,
    "name":"DIS_EVENT",
    "pollingInterval":1,
    "preNodeName":[
       "CS_PROCESS"
    ],
    "properties":[
       {
          "name":"streamName",
          "value":"dis-event"
       }
    ],
    "resources":[],
    "retryInterval":120,
    "retryTimes":0,
    "type":"DISStream"
 },
 {
    "eventTrigger":{
       "channel":"dis-event",
       "concurrent":1,
          "eventType:"DIS",
       "readPolicy":"LAST"
    },
    "failPolicy":"FAIL",
    "location":{
       "x":"848.0",
       "y":"167.0"
    },
```

```
            "maxExecutionTime":60,
            "name":"TRIP_RAW_STANDARD",
            "pollingInterval":10,
            "preNodeName":[
               "DIS_EVENT"
            ],
            "properties":[
               {
                  "name":"scriptName",
                  "value":"TRIP_RAW_STANDARD"
               },
               {
                  "name":"database",
                  "value":"lixinlong"
               },
               {
                  "name":"queueName",
                  "value":"default"
               }
            ],
            "resources":[],
            "retryInterval":120,
            "retryTimes":0,
            "type":"DLISQL"
         }
      ],
      "params":[
         {
            "name":"dis_channel",
            "value":"dis_input"
         }
      ],
      "processType":"REAL_TIME",
      "resources":[],
      "schedule":{
         "type":"EXECUTE_ONCE"
      },
      "version":"1.0"
   }
```

- Success response

  HTTP status code 204

- Failure response

  HTTP status code 400

  ```
  {
     "error_code":"DLF.0102",
     "error_msg":"The job name already exists."
  }
  ```

4.4.2 Editing a Job

Function

This API is used to edit a job.

URI

- URI format

  PUT /v1/{project_id}/jobs/{name}

- Parameter description

Table 4-89 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**.
name	Yes	String	Job name.

Request Message

For details, see **Request Message**.

Example

Modify the properties of job **dliJob1**.

- Request

```
PUT /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/dliJob1
{
   "logPath":"obs://dlf-log-b384b9e9ab9b4ee8994c8633aabc9505",
   "name":"dliJob1",
   "nodes":[
      {
         "failPolicy":"FAIL",
         "location":{
            "x":"385.0",
            "y":"150.0"
         },
         "maxExecutionTime":60,
         "name":"DIS_INPUT",
         "pollingInterval":1,
         "preNodeName":[],
         "properties":[
            {
               "name":"streamName",
               "value":"csinput"
            }
         ],
         "resources":[],
         "retryInterval":120,
         "retryTimes":0,
         "type":"DISStream"
      },
      {
         "failPolicy":"FAIL",
         "location":{
            "x":"572.0",
            "y":"151.0"
         },
         "maxExecutionTime":60,
         "name":"CS_PROCESS",
         "pollingInterval":10,
         "preNodeName":[
            "DIS_INPUT"
         ],
         "properties":[
            {
               "name":"scriptName",
               "value":"CS_PROCESS_TRIP"
            },
```

```
            {
                "name":"jobName",
                "value":"CS_PROCESS"
            },
            {
                "name":"jobType",
                "value":"flink_sql_job"
            },
            {
                "name":"spuNumber",
                "value":"2"
            },
            {
                "name":"parallelNumber",
                "value":"1"
            }
        ],
        "resources":[],
        "retryInterval":120,
        "retryTimes":0,
        "type":"CSJob"
    },
    {
        "failPolicy":"FAIL",
        "location":{
            "x":"718.0",
            "y":"121.0"
        },
        "maxExecutionTime":60,
        "name":"DIS_EVENT",
        "pollingInterval":1,
        "preNodeName":[
            "CS_PROCESS"
        ],
        "properties":[
            {
                "name":"streamName",
                "value":"dis-event"
            }
        ],
        "resources":[],
        "retryInterval":120,
        "retryTimes":0,
        "type":"DISStream"
    },
    {
        "eventTrigger":{
            "channel":"dis-event",
            "concurrent":1,
                "eventType:"DIS",
            "readPolicy":"LAST"
        },
        "failPolicy":"FAIL",
        "location":{
            "x":"848.0",
            "y":"167.0"
        },
        "maxExecutionTime":60,
        "name":"TRIP_RAW_STANDARD",
        "pollingInterval":10,
        "preNodeName":[
            "DIS_EVENT"
        ],
        "properties":[
            {
                "name":"scriptName",
                "value":"TRIP_RAW_STANDARD"
            },
            {
```

```
                "name":"database",
                "value":"lixinlong"
            },
            {
                "name":"queueName",
                "value":"default"
            }
        ],
        "resources":[],
        "retryInterval":120,
        "retryTimes":0,
        "type":"DLISQL"
    }
  ],
  "params":[
    {
        "name":"dis_channel",
        "value":"dis_input"
    }
  ],
  "processType":"REAL_TIME",
  "resources":[],
  "schedule":{
     "type":"EXECUTE_ONCE"
  },
  "version":"1.0"
}
```

- Success response

  HTTP status code 204

- Failure response

  HTTP status code 400

```
{
  "error_code":"DLF.0100",
  "error_msg":"The job does not exists."
}
```

4.4.3 Viewing a Job List

Function

This API is used to query a list of batch or real-time jobs. A maximum of 1,000 jobs can be returned for each query.

URI

- URI format

  GET /v1/{project_id}/jobs?jobType={jobType}&offset={offset}&limit={limit}&jobName={jobName}

- 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**.

Parameter	Mandatory	Type	Description
jobType	No	String	Job type. • **REAL_TIME**: real-time processing • **BATCH**: batch processing Default value: **BATCH**
offset	No	Integer	Start page of the paging list. Default value: **0** The value must be greater than or equal to **0**.
limit	No	Integer	The maximum number of records on each page. Value range: 1 to 100 Default value: 10
jobName	No	String	Name of the job.

Request Message

Table 4-90 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-91 Parameter description

Parameter	Mandatory	Type	Description
total	Yes	Integer	Number of jobs.
jobs	Yes	List<Job>	Job list. For details, see **Table 4-92**.

Table 4-92 Job data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Job name.
jobType	Yes	String	Job type. • **REAL_TIME**: real-time processing • **BATCH**: batch processing
status	Yes	String	Job status. When **jobType** is set to **REAL_TIME**, the status is as follows: • STARTING • NORMAL • EXCEPTION • STOPPING • STOPPED When **jobType** is set to **BATCH**, the status is as follows: • **SCHEDULING** • **STOPPED** • **PAUSED**
createUser	Yes	String	Job creator.
createTime	Yes	Long	Time when the job is created.
startTime	No	Long	Time when the job starts to run.
endTime	No	Long	Time when the job stops to run.
lastInstanceStatus	No	String	Most recent running status of the job instance. This parameter is available only when **jobType** is set to **BATCH**.
lastInstanceEndTime	No	Long	Time when the most recent job instance stops to run. This parameter is available only when **jobType** is set to **BATCH**.

Example

View a batch or real-time job list.

- Request

```
GET /v1/ff6b627b9d7b45b48f773be511c1a2b8/jobs?jobType=REAL_TIME
```

- Response

```
{
   "jobs": [
      {
         "createTime": 1551668561000,
         "createUser": "dlf_l00341563",
         "jobType": "REAL_TIME",
         "lastInstanceEndTime": 1551692047000,
         "lastInstanceStatus": "success",
         "name": "wangwei_stream",
         "startTime": 1551692037000
      },
      {
         "createTime": 1551109987000,
         "createUser": "dlf_l00341563",
         "jobType": "REAL_TIME",
         "lastInstanceEndTime": 1551112382000,
         "lastInstanceStatus": "success",
         "name": "job_1795",
         "startTime": 1551112367000
      }
   ],
   "total": 55
}
```

- Failure response

  HTTP status code 400

  ```
  {
     "error_code":"DLF.3051",
     "error_msg":"The request parameter is invalid."
  }
  ```

4.4.4 Viewing Job Details

Function

This API is used to view job details.

URI

- URI format

  GET /v1/{project_id}/jobs/{name}

- 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**.
name	Yes	String	Job name.

Request Message

Table 4-93 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

For details, see the response parameters for **creating a job**.

Example

View details about job **myJob**.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob
  ```
- Success response
  ```
  {
    "basicConfig": {
      "instanceTimeout": 0,
      "priority": 0
    },
    "logPath": "obs://dlf-log-b384b9e9ab9b4ee8994c8633aabc9505",
    "name": "myJob",
    "nodes": [{
      "failPolicy": "FAIL",
      "location": {
        "x": "385.0",
        "y": "150.0"
      },
      "maxExecutionTime": 60,
      "name": "DIS_INPUT",
      "pollingInterval": 1,
      "preNodeName": [],
      "properties": [{
        "name": "streamName",
        "value": "csinput"
      }],
      "resources": [],
      "retryInterval": 120,
      "retryTimes": 0,
      "type": "DISStream"
    },
    {
      "failPolicy": "FAIL",
      "location": {
        "x": "572.0",
        "y": "151.0"
      },
      "maxExecutionTime": 60,
  ```

```
    "name": "CS_PROCESS",
    "pollingInterval": 10,
    "preNodeName": ["DIS_INPUT"],
    "properties": [{
        "name": "scriptName",
        "value": "CS_PROCESS_TRIP"
    },
    {
        "name": "jobName",
        "value": "CS_PROCESS"
    },
    {
        "name": "jobType",
        "value": "flink_sql_job"
    },
    {
        "name": "clusterName",
        "value": "-1"
    },
    {
        "name": "spuNumber",
        "value": "2"
    },
    {
        "name": "parallelNumber",
        "value": "1"
    }],
    "resources": [],
    "retryInterval": 120,
    "retryTimes": 0,
    "type": "CSJob"
},
{
    "failPolicy": "FAIL",
    "location": {
        "x": "718.0",
        "y": "121.0"
    },
    "maxExecutionTime": 60,
    "name": "DIS_EVENT",
    "pollingInterval": 1,
    "preNodeName": ["CS_PROCESS"],
    "properties": [{
        "name": "streamName",
        "value": "dis-event"
    }],
    "resources": [],
    "retryInterval": 120,
    "retryTimes": 0,
    "type": "DISStream"
},
{
    "eventTrigger": {
        "channel": "dis-event",
        "concurrent": 1,
        "engineType": "DIS",
        "failPolicy": "CONTINUE",
        "readPolicy": "LAST"
    },
    "failPolicy": "FAIL",
    "location": {
        "x": "848.0",
        "y": "167.0"
    },
    "maxExecutionTime": 60,
    "name": "TRIP_RAW_STANDARD",
    "pollingInterval": 10,
    "preNodeName": ["DIS_EVENT"],
    "properties": [{
```

```
            "name": "scriptName",
            "value": "TRIP_RAW_STANDARD"
        },
        {
            "name": "database",
            "value": "lixinlong"
        },
        {
            "name": "queueName",
            "value": "default"
        }],
        "resources": [],
        "retryInterval": 120,
        "retryTimes": 0,
        "type": "DLISQL"
    }],
    "params": [{
        "name": "dis_channel",
        "value": "dis_input"
    }],
    "processType": "REAL_TIME",
    "resources": [],
    "schedule": {
        "type": "EXECUTE_ONCE"
    },
    "version": "1.0"
}
```


- Failure response

  HTTP status code 400


```
{
    "error_code":"DLF.0100",
    "error_msg":"The job does not exists."
}
```


4.4.5 Exporting a Job

Function

This API is used to export a job, including job definitions, job dependency scripts, and CDM job definitions.

URI

- URI format

  POST /v1/{project_id}/jobs/{name}/export

- Parameter description

  Table 4-94 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**.
name	Yes	String	Job name.

Request Message

Table 4-95 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-96 Parameter description

Parameter	Mandatory	Type	Description
exportDepend	No	boolean	Specifies whether to export the scripts and resources that the job depends on. Default value: true

Response Message

The value of **Content-Type** in the response message is **application/octet-stream** that needs to be parsed and converted into a file. For details, see **Parsing a Stream in a Response Message**. Response messages are compressed as a file. The file name format is **DLF**_*job_name*.**zip**. The file directory is as follows:

```
jobs
├─{job_name}.job
scripts
├─{script_name}.script
resources
├─{resource_name}.resource
```

The following table describes the file directory parameters.

Table 4-97 Parameter description

Parameter	Mandatory	Type	Description
job_name	Yes	String	Job name.
script_name	No	String	Name of the script that the job depends on.
resource_name	No	String	Name of the resource that the job depends on.

- {job_name}.job

  The parameters in the file are the same as the request parameters of the API for creating a job. For details, see **Creating a Job**.

- {script_name}.script

  The parameters in the file are the same as the request parameters of the API for creating a resource. For details, see **Creating a Script**.

Example

Export job **myJob**.

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob/export
{
   "exportDepend":true
}
```

- Success response

  The name of the exported file is **DLF_myJob.zip**. The file structure after decompression is as follows:

```
jobs
├─myJob.job
scripts
├─CS_PROCESS_TRIP.script
├─TRIP_RAW_STANDARD.script
```

- Failure response

  HTTP status code 400

```
{
   "error_code":"DLF.0100",
   "error_msg":"The job does not exists."
}
```

4.4.6 Batch Exporting Jobs

Function

This API is used to batch export jobs, including job dependency scripts and CDM job definitions.

URI

- URI format

  POST /v1/{project_id}/jobs/batch-export

- Parameter description

  Table 4-98 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**.

Request Message

Table 4-99 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Table 4-100 Parameter description

Parameter	Mandatory	Type	Description
jobList	Yes	List data structure	A list of jobs to be exported. A maximum of 100 jobs can be exported at a time.
exportDepend	No	boolean	Specifies whether to export the scripts and resources that the job depends on. Default value: **true**

Response Message

The response message of batch exporting jobs is the same as that of exporting jobs. For details, see **Response Message**.

Example

Export two jobs named **job_batch** and **job_stream**.

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/batch-export
{
   "jobList":["job_batch","job_stream"],
   "exportDepend":true
}
```

- Success response

  The name of the exported file is **jobs.zip**. The file structure after decompression is as follows:

```
jobs
+---job_batch
```

```
|       rds_sql.script
|       job_batch.job
\---job_stream
job_stream.job
```

job_batch.job and **job_stream.job** are job definition files.

dli_sql.script is the DLI SQL script file used by **job_batch.job**.

- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.3051",
    "error_msg":"The request parameter is invalid."
}
```

4.4.7 Viewing a Job File

Function

This API is used to check whether there are jobs and scripts in the job file to be imported from OBS to DLF.

URI

- URI format

  POST /v1/{project_id}/jobs/check-file

- Parameter description

Table 4-101 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**.

Request Message

Table 4-102 Parameter description

Parameter	Mandatory	Type	Description
path	Yes	String	If OBS is deployed, the job definition file is stored on OBS, for example, obs://myBucket/jobs.zip. If OBS is not deployed deployed, the job definition file is stored in the local PC.

Response Message

Table 4-103 Parameter description

Parameter	Mandatory	Type	Description
jobs	No	List<Job>	Job information. For details, see **Table 4-104**.
scripts	No	List<Script>	Script information. For details, see **Table 4-105**.
resources	No	List<Resource>	Resource information referenced in the job. For details, see **Table 4-106**.

Table 4-104 Data structure description of **jobs**

Parameter	Mandatory	Type	Description
params	No	Map<String,String>	Job parameter.
name	Yes	String	Job name.
path	Yes	String	Path of the job.

Table 4-105 Data structure description of **Script**

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the script.
path	Yes	String	Path of the script.

Table 4-106 Data structure description of **Resource**

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the resource.
status	Yes	Int	Resource status. **0** indicates that the resource does not exist; **1** indicates that the resource exists.

Parameter	Mandatory	Type	Description
type	Yes	String	Resource type. The value can be: • **DWS_CLUSTER** • **DWS_CONNECTION** • **DLI_QUEUE** • **DLI_CONNECTION** • **DIS_STREAM** • **CDM_CLUSTER** • **GES_GRAPH**

Example

View the parameter definitions in the job file on OBS.

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/check-file
{
"path": "obs://aaaaa/DLF_myJob.zip"
}
```

- Success response

```
{
   "jobs":[
      {
         "name":"test",
         "path":"/test",
         "params":{
            "ddd":"dddd"
         }
      },
      {
         "name":"test1",
         "path":"/test",
         "params":{
            "ddd":"dddd"
         }
      }
   ],
   "scripts":[
      {
         "name":"script1",
         "path":"/path1"
      },
      {
         "name":"script2",
         "path":"/path1"
      }
   ]
}
```

- Failure response

  HTTP status code 400

```
{
   "error_code":"DLF.0815",
   "error_msg":"Fail to read OBS file."
}
```

Status Code

For details about status codes, see **Status Codes**.

4.4.8 Importing a Job

Function

This API is used to import one or more job files from OBS to DLF.

NOTE

Before using this API, store job files in OBS buckets.

URI

- URI format

  POST /v1/{project_id}/jobs/import
- Parameter description

Table 4-107 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**.

Request Message

Table 4-108 Parameter description

Parameter	Mandatory	Type	Description
path	Yes	String	If OBS is deployed, it refers to the OBS path for storing the job definition file. For the format of the job definition file, see the response message of the exported job, for example, obs://myBucket/jobs.zip. If OBS is not deployed deployed, the job definition file is stored in the local PC.
params	No	Map<String,String>	Public job parameter.

Parameter	Mandatory	Type	Description
sameNamePolicy	No	String	Policy for specifying how to handle duplicate names. The options are as follows: • **SKIP** • **OVERWRITE** The default value is **SKIP**.
jobsParam	No	List<JobParam>	Job parameter. For details, see **Table 4-109**.
executeUser	No	String	User that executes the job.

Table 4-109 Parameter description of **JobParam**

Parameter	Mandatory	Type	Description
name	Yes	String	Job name.
params	No	Map<String,String>	Job parameter.

Response Parameter

Table 4-110 Parameter description

Parameter	Mandatory	Type	Description
taskId	Yes	String	ID of the task. Used to call the API for querying system tasks to obtain the import status.

Example

Import jobs from OBS to DLF. If there are jobs and scripts with the same name, overwrite them.

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/import
{
"path": "obs://aaaaa/job_batch.zip",
"sameNamePolicy": "OVERWRITE",
"jobsParam": [{
"name": "job_batch",
"params": {
"streamName": "dis-AHTr"
```

```
}
}]
}
```

- Success response

  HTTP status code 200

```
{
"taskId":"008aae2e675933c7016759418e870000"
}
```

- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.0815",
    "error_msg":"Fail to read OBS file."
}
```

4.4.9 Starting a Job

Function

This API is used to start a job.

URI

- URI format

  POST /v1/{project_id}/jobs/{name}/start

- Parameter description

  Table 4-111 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**.
name	Yes	String	Job name.

Request Message

Table 4-112 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Parameter description

Parameter	Mandatory	Type	Description
jobParams	No	List<JobParam>	Parameter for starting the job.

JobParam data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the parameter. It cannot exceed 64 characters.
value	Yes	String	Value of the parameter. It cannot exceed 1,024 characters.

Response Message

None.

Example

Start job **myJob**.

- Request
  ```
  POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob/start
  ```
- Success response

  HTTP status code 204
- Failure response

  HTTP status code 400

```
{
    "error_code":"DLF.0100",
    "error_msg":"The job does not exists."
}
```

4.4.10 Stopping a Job

Function

This API is used to stop a job.

URI

- URI format
  POST /v1/{project_id}/jobs/{name}/stop
- Parameter description

Table 4-113 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**.
name	Yes	String	Job name.

Request Message

Table 4-114 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

None.

Example

Stops job **myJob**.

- Request
  ```
  POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob/stop
  ```
- Success response

  HTTP status code 204
- Failure response

  HTTP status code 400

  ```
  {
      "error_code":"DLF.0100",
      "error_msg":"The job does not exists."
  }
  ```

4.4.11 Deleting a Job

Function

This API is used to delete a job.

URI

- URI format

  DELETE /v1/{project_id}/jobs/{name}
- Parameter description

Table 4-115 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**.
name	Yes	String	Job name.

Request Message

Table 4-116 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

None.

Example

- Request
  ```
  DELETE /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/joba
  ```
- Response

  HTTP status code 204
- Failure response

  HTTP status code 400

  ```
  {
     "error_code":"DLF.0100",
     "error_msg":"The job does not exists."
  }
  ```

Status Code

For details about status codes, see **Status Codes**.

4.4.12 Executing a Job Immediately

Function

This API is used to execute a job immediately and check whether the job can be executed successfully.

URI

- URI format

  POST /v1/{project_id}/jobs/{name}/run-immediate
- Parameter description

  Table 4-117 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**.
name	Yes	String	Name of the job.

Request Message

Table 4-118 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Parameter description

Parameter	Mandatory	Type	Description
jobParams	No	List<JobParam>	Parameter for starting the job.

JobParam data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the parameter. It cannot exceed 64 characters.
value	Yes	String	Value of the parameter. It cannot exceed 1,024 characters.

Response Message

Table 4-119 Parameter description

Parameter	Mandatory	Type	Description
instanceId	Yes	Long	Job instance ID.

Example

Execute the **myJob** job once.

- Request

```
POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/myJob/run-immediate
{
   "jobParams":[
      {
         "name":"aaa",
         "value":"111"
      },
      {
         "name":"bbb",
         "value":"222"
      }
   ]
}
```

- Success response

```
{
   "instanceId":132343
}
```

- Failure response

  HTTP status code 400

```
{
   "error_code":"DLF.0100",
   "error_msg":"The job does not exists."
}
```

4.4.13 Viewing Running Status of a Real-Time Job

Function

This API is used to view running status of a real-time job.

URI

- URI format

  GET /v1/{project_id}/jobs/{name}/status

- Parameter description

Table 4-120 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**.
name	Yes	String	Job name.

Request Message

Table 4-121 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-122 Parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the solution.
nodes	No	List data structure	Node status list.
status	No	String	Job status. The value can be: • **STARTING** • **NORMAL** • **EXCEPTION** • **STOPPING** • **STOPPED**
startTime	Yes	Date	Start time.
endTime	No	Date	End time.
lastUpdateTime	No	Date	Last update time.

Table 4-123 Data structure description of **nodes**

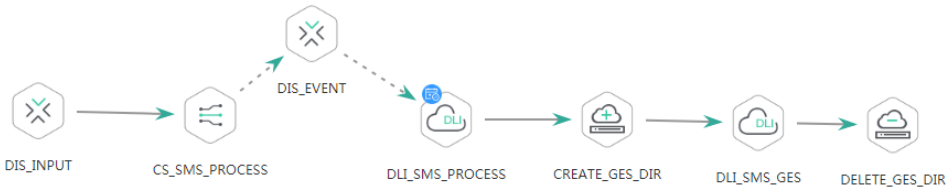
Parameter	Mandatory	Type	Description
name	Yes	String	Node name.

Parameter	Mandatory	Type	Description
status	No	String	Node status. The value can be: • **STARTING** • **NORMAL** • **EXCEPTION** • **STOPPING** • **STOPPED**
logPath	No	String	Path for storing node run logs.
type	Yes	String	Node type. • **Hive SQL**: Runs Hive SQL scripts. • **Spark SQL**: Runs Spark SQL scripts. • **DWS SQL**: Runs DWS SQL scripts. • **DLI SQL**: Runs DLI SQL scripts. • **Shell**: Runs shell SQL scripts. • **CDM Job**: Runs CDM jobs. • **DIS Transfer Task:** Creates DIS dump tasks. • **CS Job**: Creates and starts CloudStream jobs. • **CloudTable Manager**: Manages CloudTable tables, including creating and deleting tables. • **OBS Manager**: Manages OBS paths, including creating and deleting paths. • **REST API:** Sends REST API requests. • **SMN**: Sends short messages or emails. • **MRS Spark**: Runs Spark jobs of MRS. • MapReduce: Runs MapReduce jobs of MRS.

Example

View running status of real-time job **job_sms** and the running status of each node in the job. **job_sms** has seven nodes, as shown in the following figure:

Figure 4-2 job_sms



- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/job_sms/status
  ```
- Success response (The **DLI_SMS_GES** node runs abnormally during job running.)
  ```
  {
     "lastUpdateTime": 1551409406000,
     "name": "job_sms",
     "nodes": [{
        "name": "CREATE_GES_DIR",
        "status": "NORMAL",
        "totalGetBytes": 0,
        "totalGetRecords": 0,
        "totalPutBytes": 0,
        "totalPutRecords": 0,
        "type": "Create OBS"
     },
     {
        "logPath": "obs://dlf-log-b384b9e9ab9b4ee8994c8633aabc9505/job_sms/2019-03-01
  11_04_24.000/CS_SMS_PROCESS/CS_SMS_PROCESS-IrU8hS82_20190301_110433_183.job",
        "name": "CS_SMS_PROCESS",
        "runningData": "{\"jobName\":\"CS_SMS_PROCESS\",\"jobId\":\"107016\",\"App_dis-input\":
  \"app_sms_process\"}",
        "status": "NORMAL",
        "totalGetBytes": 0,
        "totalGetRecords": 165,
        "totalPutBytes": 0,
        "totalPutRecords": 165,
        "type": "ExecuteCloudStream"
     },
     {
        "name": "DELETE_GES_DIR",
        "status": "STOPPED",
        "totalGetBytes": 0,
        "totalGetRecords": 0,
        "totalPutBytes": 0,
        "totalPutRecords": 0,
        "type": "Delete OBS"
     },
     {
        "logPath": "obs://dlf-log-b384b9e9ab9b4ee8994c8633aabc9505/job_sms/2019-03-01
  11_04_24.000/DIS_EVENT/DIS_EVENT-kefeNV5B_20190301_110439_984.job",
        "name": "DIS_EVENT",
        "runningData": "{\"app\":\"DLF_job_sms_DLI_SMS_PROCESS\",\"streamName\":\"dis-event\"}",
        "status": "NORMAL",
        "totalGetBytes": 0,
        "totalGetRecords": 8602,
        "totalPutBytes": 0,
        "totalPutRecords": 8596,
        "type": "ExecuteDISStream"
  ```

```
    },
    {
        "logPath": "obs://dlf-log-b384b9e9ab9b4ee8994c8633aabc9505/job_sms/2019-03-01
11_04_24.000/DIS_INPUT/DIS_INPUT-T18JOYTc_20190301_110428_754.job",
        "name": "DIS_INPUT",
        "runningData": "{\"streamName\":\"dis-input\"}",
        "status": "NORMAL",
        "totalGetBytes": 0,
        "totalGetRecords": 70341,
        "totalPutBytes": 0,
        "totalPutRecords": 70341,
        "type": "ExecuteDISStream"
    },
    {
        "name": "DLI_SMS_GES",
        "status": "EXCEPTION",
        "totalGetBytes": 0,
        "totalGetRecords": 0,
        "totalPutBytes": 0,
        "totalPutRecords": 0,
        "type": "DLI SQL"
    },
    {
        "name": "DLI_SMS_PROCESS",
        "status": "NORMAL",
        "totalGetBytes": 0,
        "totalGetRecords": 208,
        "totalPutBytes": 0,
        "totalPutRecords": 208,
        "type": "DLI SQL"
    }],
    "startTime": 1551409465000,
    "status": "NORMAL"
}
```


- Failure response

  HTTP status code 400


```
{
    "error_code":"DLF.0100",
    "error_msg":"The job does not exists."
}
```


4.4.14 Viewing a Job Instance List

Function

This API is used to view a job instance list.

A job instance is generated each time you run a batch job for which periodic scheduling or event-based scheduling is configured. If a real-time job contains a node for which periodic scheduling or event-based scheduling is configured, you can call this API to view the instance list of the subjobs associated with the node. The format of the **jobName** parameter is *real-time job name_node name*.

URI

- URI format

  GET /v1/{project_id}/jobs/instances/detail?jobName={jobName}&minPlanTime={minPlanTime}&maxPlanTime={maxPlanTime}&limit={limit}&offset={offset}&status={status}

- Parameter description

Table 4-124 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**.
jobName	No	String	Job name. • If you want to query the instance list of a specific batch job, jobName is the batch job name. • If you want to query sub-jobs associated with a node in a real-time job, the jobName format is *real-time job name _ node name*.
minPlanTime	No	Long	The job instance whose planned execution time is later than minPlanTime is returned.
maxPlanTime	No	Long	The job instance whose planned execution time is earlier than minPlanTime is returned.
offset	No	int	Start page of the paging list. Default value: **0** The value must be greater than or equal to **0**.
limit	No	int	Specifies the maximum number of records on each page. The parameter value ranges from 1 to 1,000. Default value: 10
status	No	String	Job instance status. The value can be: • **waiting** • **running** • **success** • **fail** • **running-exception** • **pause** • **manual-stop**

Request Message

Table 4-125 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-126 Parameter description

Parameter	Mandatory	Type	Description
total	Yes	int	Total number of records.
instances	Yes	List<Instance>	Job instance status. For details, see **Table 4-127**.

Table 4-127 Instance parameter description

Parameter	Mandatory	Type	Description
jobName	Yes	String	Job name. When you view the instance list of a specified batch job, **jobName** is the name of the batch job. When you view the subjobs associated with a node in a real-time job, **jobName** is in format of *real-time job name_node name*.

Parameter	Mandatory	Type	Description
status	Yes	String	Job instance status. The value can be: • **waiting** • **running** • **success** • **fail** • **running-exception** • **pause** • **manual-stop**
planTime	Yes	Long	Planned execution time of the job instance.
startTime	Yes	Long	Actual execution start time of the job instance.
endTime	No	Long	Actual execution end time of the job instance.
executeTime	No	Long	Execution duration in milliseconds.
instanceId	Yes	Long	Job instance ID.
submitTime	Yes	Long	Time when a job is submitted.

Example 1

View the instance list of batch job **job_batch**.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/instances/detail?jobName=job_batch
  ```
- Success response
  ```
  {
     "total": 2,
     "instances": [{
        "endTime": 1551671598000,
        "executeTime": 0.3,
        "instanceId": 34765,

        "jobName": "job_batch",
        "planTime": 1551671580000,
        "startTime": 1551671580000,
        "status": "success",
        "submitTime": 1550910278706
     },
     {
        "endTime": 1551671538000,
        "executeTime": 0.3,
        "instanceId": 34764,

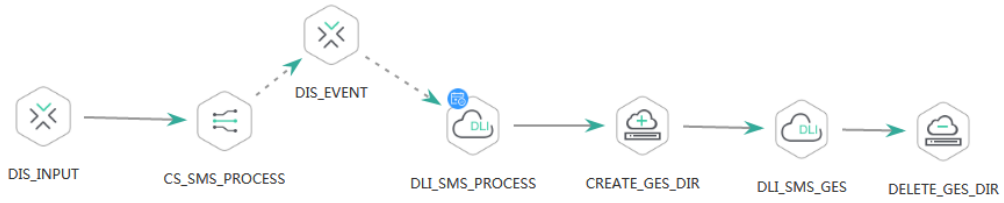
        "jobName": "job_batch",
        "planTime": 1551671520000,
        "startTime": 1551671521000,
  ```

```
    "status": "success",
    "submitTime": 1550910278706
  }]
}
```

Example 3

View the instance list of the subjobs associated with node **DLI_SMS_PROCESS** in real-time job **job_sms**. The following figure shows the **job_sms** job. Event-based scheduling is configured for the **DLI_SMS_PROCESS** node.

Figure 4-3 job_sms



- Request

  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/instances/detail?
  jobName=job_sms_DLI_SMS_PROCESS
  ```

- Response

  ```
  {
    "total": 2,
    "instances": [{
      "endTime": 1551425387000,
      "exeTime": 1.0,
      "instanceId": 30654,
      "jobName": "job_sms_DLI_SMS_PROCESS",
      "planTime": 1551425326540,
      "startTime": 1551425327000,
      "status": "fail",
      "submitTime": 1551409464657
    },
    {
      "endTime": 1551409570000,
      "exeTime": 0.5,
      "instanceId": 28960,
      "jobName": "job_sms_DLI_SMS_PROCESS",
      "planTime": 1551323851910,
      "startTime": 1551409540000,
      "status": "fail",
      "submitTime": 1551323793766
    }]
  }
  ```

  - Failure response

    HTTP status code 400

    ```
    {
      "error_code":"DLF.3051",
      "error_msg":"The request parameter is invalid."
    }
    ```

4.4.15 Viewing Job Instance Details

Function

This API is used to view job instance details, including the execution information about each node in a job instance.

URI

- URI format

  GET /v1/{project_id}/jobs/{jobName}/instances/{instanceId}

- Parameter description

  Table 4-128 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**.
jobName	Yes	String	Job name.
instanceId	Yes	Long	Job instance ID.

Request Message

Table 4-129 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-130 Parameter description

Parameter	Mandatory	Type	Description
jobName	Yes	String	Job name.

Parameter	Mandatory	Type	Description
instanceId	Yes	Long	Job instance ID.
status	Yes	String	Job instance status. The value can be: • **waiting** • **running** • **success** • **fail** • **running-exception** • **pause** • **manual-stop**
planTime	Yes	Long	Planned execution time of the job instance.
startTime	Yes	Long	Actual execution start time of the job instance.
endTime	No	Long	Actual execution end time of the job instance.
executeTime	No	Long	Execution duration in milliseconds.
total	Yes	int	Total number of node records.
nodes	Yes	List<Node>	Node instance status. For details, see **Table 4-131**.

Table 4-131 Node parameter description

Parameter	Mandatory	Type	Description
nodeName	Yes	String	Node name.
status	Yes	String	Node status. The value can be: • **waiting** • **running** • **success** • **fail** • **skip** • **pause** • **manual-stop**
planTime	Yes	Long	Planned execution time of the job instance.

Parameter	Mandatory	Type	Description
startTime	Yes	Long	Actual execution start time of the node.
endTime	No	Long	Actual execution end time of the node.
type	Yes	String	Node type.
retryTimes	No	Int	Number of attempts upon a failure.
instanceId	Yes	Long	Job instance ID.
inputRowCount	No	Long	Rows of input data.
speed	No	double	Write speed (row/second)
logPath	No	String	Path for storing node execution logs.

Example

View details about the instance whose ID is 34765 in job **job_batch**.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/job_batch/instances/34765
  ```
- Response
  ```
  {
     "jobName":"job_batch",
     "instanceId":34765,

     "status":"fail",
     "planTime":1551425326540,
     "startTime":1551425327000,
     "endTime":1551425387000,
     "executeTime":1,
     "instanceId":30654,
     "total":2,
     "nodes":[
        {
           "endTime":1551671590000,
           "inputRowCount":0,
           "instanceId":34765,
           "nodeName":"Dummy_8556",

           "planTime":1551671580000,
           "retryTimes":0,
           "startTime":1551671584000,
           "status":"success",
           "submitTime":1550910278706,
           "type":"Dummy"
        },
        {
           "endTime":1551671598000,
           "inputRowCount":0,
           "instanceId":34765,
           "logPath":"obs://dlf-log-b384b9e9ab9b4ee8994c8633aabc9505/job_batch/2019-03-04
  11_53_00.000/error/error.job",
  ```

```
            "nodeName":"error",

            "planTime":1551671580000,
            "retryTimes":0,
            "startTime":1551671594000,
            "status":"success",
            "submitTime":1550910278706,
            "type":"RDS SQL"
        }
    ]
}
```


- Failure response

  HTTP status code 400


```
{
    "error_code":"DLF.0137",
    "error_msg":"Job instance does not exist."
}
```


4.4.16 Stopping a Job Instance

Function

This API is used to stop a specific job instance. A job instance can be stopped only when it is in the running state.

URI

- URI format

  POST /v1/{project_id}/jobs/{jobName}/instances/{instanceId}/stop

- Parameter description

  Table 4-132 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**.
jobName	Yes	String	Name of the job.
instanceId	Yes	Long	Job instance ID.

Request Message

Table 4-133 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. ● If this parameter is not set, data in the **default** workspace is queried by default. ● If you need to query data in other workspaces, this header must be carried.

Response Message

None.

Example

- Request
  ```
  POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/job_batch/instances/34765/stop
  ```
- Success response

  HTTP status code 204
  - Failure response

    HTTP status code 400

    ```
    {
       "error_code":"DLF.0137",
       "error_msg":"Job instance does not exist."
    }
    ```

4.4.17 Retry a Job Instance

Function

This API is used to retry a specific job instance. A job instance can be retried only when it is in the successful, failed, or canceled state.

URI

- URI format

  POST /v1/{project_id}/jobs/{jobName}/instances/{instanceId}/restart
- Parameter description

Table 4-134 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**.
jobName	Yes	String	Name of the job.
instanceId	Yes	Long	Job instance ID.

Request Message

Table 4-135 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

None.

Example

- Request
  ```
  POST /v1/b384b9e9ab9b4ee8994c8633aabc9505/jobs/job_batch/instances/34765/restart
  ```
- Response

  HTTP status code 204

  – Failure response

  HTTP status code 400

  ```
  {
      "error_code":"DLF.0137",
      "error_msg":"Job instance does not exist."
  }
  ```

4.4.18 Querying a System Task

Function

This API is used to query details about asynchronous tasks.

URI

- URI format
  GET /v1/{project_id}/system-tasks/{task_id}
- Parameter description

Table 4-136 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**.
task_id	Yes	String	ID of the task.

Request Message

Table 4-137 Request header parameters

Parameter	Mandatory	Type	Description
workspace	No	String	Workspace ID. • If this parameter is not set, data in the **default** workspace is queried by default. • If you need to query data in other workspaces, this header must be carried.

Response Message

Table 4-138 Parameter description

Parameter	Mandatory	Type	Description
id	Yes	String	ID of the task.
name	Yes	String	Name of the task.
startTime	Yes	Long	Time when the task starts to run.
endTime	No	Long	Time when the task stops to run.
lastUpdate	Yes	Long	Time when the task is updated.

Parameter	Mandatory	Type	Description
status	Yes	String	Task status. The value can be: • **RUNNING** • **SUCCESSFUL** • **FAILED**
message	No	String	Task information.
subtasks	No	List<SubTask>	Subtask. For details, see **Table 4-139**.

Table 4-139 SubTask parameter description

Parameter	Mandatory	Type	Description
id	Yes	String	ID of the subtask.
name	Yes	String	Name of the subtask.
startTime	Yes	Long	Time when the subtask starts to run.
endTime	No	Long	Time when the subtask stops to run.
lastUpdate	Yes	Long	Time when the subtask is updated.
status	Yes	String	Subtask status. The value can be: • **RUNNING** • **SUCCESSFUL** • **FAILED**

Example 1

After the solution is imported successfully, query the task information.

- Request
  ```
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/system-tasks/008aae2e675d3bcb01675d3c66f20000
  ```
- Response
  ```
  {
  {
  "endTime":1543457514000,
  "id":"008aae2e675d3bcb01675d3c66f20000",
  "lastUpdate":1543457514000,
  "name": "Import the solution from OBS path obs://aaaaa/traffic.zip.",
  "projectId":"b384b9e9ab9b4ee8994c8633aabc9505",
  "startTime":1543457499000,
  "status":"SUCCESSFUL",
  "subtasks":[
  {
  ```

"id":"008aae2e675d3bcb01675d3c9f5f0002",
"lastUpdate":1543457513000,
"name": "Import the **job_batch** job.",
"status":"SUCCESSFUL",
"taskId":"008aae2e675d3bcb01675d3c66f20000"
},
{
"id":"008aae2e675d3bcb01675d3c9fdc0003",
"lastUpdate":1543457513000,
"name": "Import the **job_stream** job.",
"status":"SUCCESSFUL",
"taskId":"008aae2e675d3bcb01675d3c66f20000"
}
]
}

Example 2

After the solution fails to be imported, query the task information.

- Request
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/systemTasks/008aae2e675d3bcb01675d3e6b960004
- Response
{
"endTime":1543457631000,
"id":"008aae2e675d3bcb01675d3e6b960004",
"lastUpdate":1543457631000,
"message": "Failed to read the OBS file **obs://aaaaa/traffic2.zip**. "
"name": "Import the solution from OBS path **obs://aaaaa/traffic2.zip**.",
"projectId":"b384b9e9ab9b4ee8994c8633aabc9505",
"startTime":1543457631000,
"status":"FAILED",
"subtasks":[]
}

Example 3

Query the task for starting a solution.

- Request
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/systemTasks/008aae2e675d3bcb01675d40ab3f0005
- Response
{
"endTime":1543457794000,
"id":"008aae2e675d3bcb01675d40ab3f0005",
"lastUpdate":1543457794000,
"name":"Start the **traffic** solution",
"projectId":"b384b9e9ab9b4ee8994c8633aabc9505",
"startTime":1543457778000,
"status":"FAILED",
"subtasks":[
{
"id":"008aae2e675d3bcb01675d40e8560006",
"lastUpdate":1543457794000,
"name": "Start the **job_batch** job",
"status":"FAILED",
"taskId":"008aae2e675d3bcb01675d40ab3f0005"
}
]
}

Example 4

Query the task for exporting a job.

- Request
  GET /v1/b384b9e9ab9b4ee8994c8633aabc9505/systemTasks/008aae2e675d3bcb01675d40ab3f0005
- Response
  ```
  {
  "endTime": 1544777083000,
  "id": "008aae2e67abe2ff0167abe3a28f0000",
  "lastUpdate": 1544777083000,
  "name": "Import the job from obs://aaaaa/job_batch.zip.",
  "projectId": "b384b9e9ab9b4ee8994c8633aabc9505",
  "startTime": 1544777081000,
  "status": "SUCCESSFUL",
  "subtasks": [{
  "id": "008aae2e67abe2ff0167abe3a7e70002",
  "lastUpdate": 1544777083000,
  "name": "Import the job_batch job.",
  "status": "SUCCESSFUL",
  "taskId": "008aae2e67abe2ff0167abe3a28f0000"
  }]
  }
  ```
- Failure response

  HTTP status code 400

  ```
  {
     "error_code":"DLF.0810",
     "error_msg":"Task does not exist"
  }
  ```

4.5 Data Structure

Table 4-140 describes common request headers.

Table 4-140 Common request header fields

Parameter	Mandatory	Type	Description
X-Sdk-Date	Yes	String	Time when the request is sent. The time is in **YYYYMMDD'T'HHMMSS'Z'** format. The value is the current GMT time of the system.
Authorization	Yes	String	Authentication information. The value can be obtained from the request signing result. For details, see **AK/SK-based Authentication**.
Host	Yes	String	Server information of the resource 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.

Parameter	Mandatory	Type	Description
Content-type	Yes	String	Request body MIME type.
Content-Length	No	Int	Size of the request body, measured in bytes. Mandatory for POST and PUT requests, but must be left blank for GET requests.
X-Project-Id	No	String	Project ID used to obtain a token for each project.
X-Auth-Token	No	String	User token. This parameter is mandatory for authentication using tokens.

Table 4-141 describes common response headers.

Table 4-141 Common response headers

Parameter	Type	Description
Content-Length	Int	Length of the response body. The unit is byte.
Date	String	Time when a response is returned.
Content-type	String	Request body MIME type.

5 Appendix

5.1 Status Codes

A status code consists of three digits. The first digit defines the class of a response. There are five values for the first digit:

- 1*xx*: indication information, indicating that the request has been received and can be further processed.
- 2*xx*: success, indicating that the request has been received, understood, and accepted.
- 3*xx*: redirection, indicating that the request requires further operations before it can be completed.
- 4*xx*: client error, indicating that there is a syntax error in the request or the request cannot be implemented.
- 5*xx*: server error, indicating that the server has failed to implement a valid request.

Table 5-1 describes status codes.

Table 5-1 Status codes

Status Code	Message	Description
100	Continue	The client should continue with its request. This interim response is used to inform the client that part of the request has been received and has not yet been rejected by the server.
101	Switching Protocols	The protocol should be switched. The protocol can only be switched to a newer protocol. For example, the current HTTP protocol is switched to a later version of HTTP.
200	OK	The request has been fulfilled.

Status Code	Message	Description
201	Created	The request has been fulfilled and a new resource has been created.
202	Accepted	The request has been accepted, but the processing has not been completed.
203	Non-Authoritative Information	The server has successfully processed the request, but is returning information that may be from another source.
204	NoContent	The request has been fulfilled, but the HTTP response does not contain a response body. The status code is returned in response to an HTTPS 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 the partial 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	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 is available only through a proxy.
306	Unused	The HTTP status code is no longer used.
400	BadRequest	The request is invalid. The client should not repeat the request without modifications.

Status Code	Message	Description
401	Unauthorized	This 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.
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 server has timed out waiting for the request. The client may repeat the request without modifications at a later time.
409	Conflict	The request could not be processed due to a conflict with the current state of the resource. This status code indicates that the resource that the client is attempting to create already exists, or that the request has failed to be processed because of the update of the conflict request.
410	Gone	The requested resource has been deleted permanently and is no longer available.
411	Length Required	The server is refusing to process the request without a defined **Content-Length**.

Status Code	Message	Description
412	Precondition Failed	The server did not meet one of the preconditions that the requester put on the request.
413	Request Entity Too Large	The server is refusing to process a request because the request entity is too large for the server to process. The server may close the connection to prevent the client from continuing the request. If the server is only temporarily unable to process the request, the response will contain a **Retry-After** header field.
414	Request-URI Too Large	The Request-URI is 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 has failed to meet the requirements of the **Expect** request-header field.
422	UnprocessableEntity	The request is well-formed but cannot be processed due to semantic errors.
429	TooManyRequests	The client has sent excessive number of requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server has received an excessive number of requests within a given time (beyond its processing capability). In this case, the client should resend the request after the time specified in the **Retry-After** header of the response has elapsed.
500	InternalServerError	The server is able to receive the request but unable to understand it.
501	Not Implemented	The server does not support the function required to fulfill the request.
502	Bad Gateway	The server was acting as a gateway or proxy and received an invalid request from the remote server.
503	ServiceUnavailable	The requested service is invalid. The client should not repeat the request without modifications.

Status Code	Message	Description
504	ServerTimeout	The request cannot be fulfilled within a given time. This status code is returned to the client only if 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.

5.2 Error Codes

The error response is in the following format:

{

"error_code":"DLF.1001",

"error_msg":"The job not found"

}

Table 5-2 describes the error codes.

Table 5-2 Error codes

Error Code	Error Message
DLF.0100	The job does not exist.
DLF.0101	The job quota has reached the upper limit.
DLF.0102	The job name has been used by another job.
DLF.0136	This API can be called only for batch jobs.
DLF.0137	The job instance does not exist.
DLF.0201	The script fails to be created.
DLF.0202	The script name has been used by another script.
DLF.0203	The script fails to be modified.
DLF.0802	The specified parameter is not configured.
DLF.0803	The OBS path is invalid.
DLF.0810	The queried task does not exist.
DLF.0815	The OBS file fails to be accessed.
DLF.1006	The job node is empty.
DLF.1242	The OBS bucket does not exist.
DLF.3004	The job name cannot be left blank.

Error Code	Error Message
DLF.3018	The job name is invalid.
DLF.3025	The job scheduling parameter is invalid.
DLF.3050	The job description is invalid.
DLF.6201	The script does not exist.
DLF.6205	The script execution instance does not exist.
DLF.6241	The resource to be queried does not exist.
DLF.6247	The script type is not specified.
DLF.6253	The script quota has reached the upper limit.
DLF.6258	The directory contains data that cannot be deleted.
DLF.6259	A resource with the same name already exists in the directory.
DLF.6263	The resource type is invalid.
DLF.6264	The OBS path of the resource file is invalid.
DLF.6265	The resource description is invalid.
DLF.6271	Each user can create only one DLI connection.
DLF.6309	The connection name has been used by another connection.
DLF.6322	The data connection does not exist.
DLF.6323	The connection type cannot be modified.
DLF.6416	The cluster has been occupied by another connection.
DLF.6418	Either the DWS cluster name or the access address and port number of the cluster must be configured.

5.3 Obtaining a Project ID

Obtaining a Project ID by Calling an API

You can obtain the project ID by calling the API used to **query project information based on the specified criteria**.

The API used to obtain a project ID is **GET https://{Endpoint}/v3/projects/**. ***Endpoint*** indicates the IAM endpoint, which can be obtained from **Regions and Endpoints**. For details about API authentication, see **Authentication**.

The following is an example response. The value of **id** is the project ID. If multiple IDs are returned, obtain the desired one based on your actual region (name).

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "region-name",
            "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"
    }
}
```

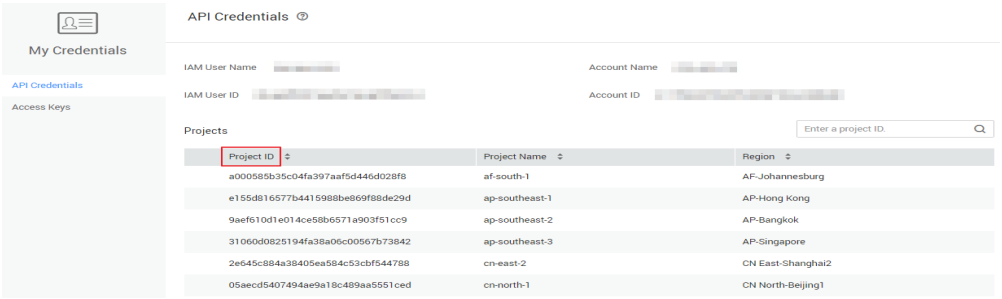

Obtaining a Project ID from the Console

A project ID is required for some URLs 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 **API Credentials** page, view project IDs in the project list.

Figure 5-1 Viewing project IDs



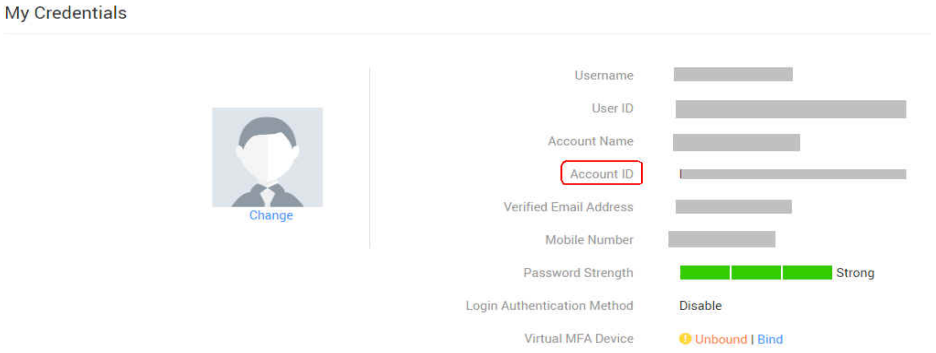
5.4 Obtaining an Account ID

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

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

   On the **My Credentials** page, view the account ID.

Figure 5-2 Obtaining an account ID

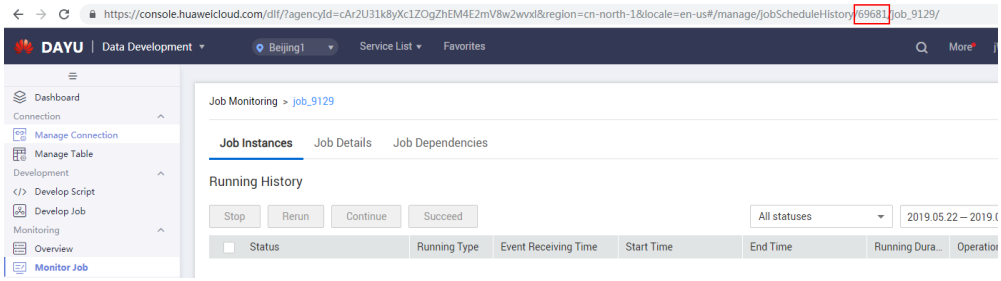


5.5 Obtaining a Job ID

A job ID (**jobid**) is required for some URLs when an API is called. To obtain a job ID, perform the following operations:

1. Log in to the management console.
2. In the navigation tree of the DLF console, choose **Monitoring** > **Monitor Job**.
3. On the job monitoring page, click a job name and view the job name in the URL.

Figure 5-3 Obtaining a job ID



5.6 Parsing a Stream in a Response Message

The response messages of the job export API and connection export API are streams that need to be converted to files. For details, see the following sample code:

```
String EXPORT_JOB_URL = "https://{endpoint}/v1/{project_id}/jobs/{job_name}/export";

try (CloseableHttpClient httpClient = HttpClients.createDefault()) {
    HttpPost httpPost = new HttpPost(EXPORT_JOB_URL);
    httpPost.setHeader("Content-Type", "application/json; charset=UTF-8");
    httpPost.setHeader("Accept", "application/octet-stream");
    httpPost.setHeader("X-Auth-Token", token);

    HttpResponse response = httpClient.execute(httpPost);
    int statusCode = response.getStatusLine().getStatusCode();
    if (statusCode == 200) {
        String filePath = "d:";
        String fileName = "job.zip";
        FileOutputStream fileOutputStream = new FileOutputStream(filePath + fileName);
        response.getEntity().writeTo(fileOutputStream);
```

```
    } else {
        System.out.println(statusCode);
    }
}
```

A Change History

Release Date	Description
2020-04-28	This issue is the first official release.