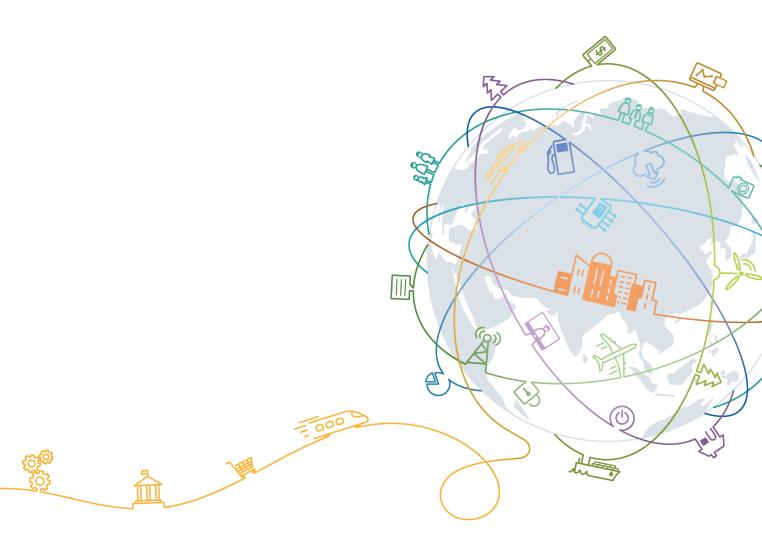
Data Lake Factory

API Reference

Issue 01

Date 2020-12-08





Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

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

Notice

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

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

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base

Bantian, Longgang Shenzhen 518129

People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Contents

1 Before You Start	1
1.1 Overview	1
1.2 API Calling	1
1.3 Endpoints	1
1.4 Constraints	2
1.5 Concepts	2
2 API Overview	4
3 Calling APIs	5
3.1 Making an API Request	5
3.2 Authentication	10
3.3 Response	11
4 Getting Started	13
5 DLF APIs	15
5.1 Job Development APIs	15
5.1.1 Running a Job	15
5.1.2 Stopping Scheduling a Job	16
6 Data Structure	18
7 Permissions Policies and Supported Actions	20
7.1 Introduction	20
8 Appendix	29
8.1 Status Codes	29
8.2 Error Codes	33
8.3 Obtaining a Project ID	34
8.4 Obtaining an Account ID	
8.5 Obtaining a Job ID	
8.6 Parsing a Stream in a Response Message	37
A Change History	38

Before You Start

1.1 Overview

Welcome to *Data Lake Factory API Reference*. Data Lake Factory (DLF) is a HUAWEI CLOUD platform product. It provides a one-stop big data collaborative development environment with fully managed data scheduling capabilities. DLF supports batch processing, real-time pipeline orchestration, monitoring, O&M, and graphical programming (that reduces SQL statements by 80%), helping users quickly build big data processing centers.

This document describes how to use application programming interfaces (APIs) to perform operations on DLF, such as performing or stopping jobs. For details about all supported operations, see API Overview.

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

1.2 API Calling

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

1.3 Endpoints

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

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

Region	Endpoint Region	Endpoint	Protocol	
CN-Hong Kong	ap-southeast-1	dlf.ap- southeast-1.myhu aweicloud.com	HTTPS	
AP-Bangkok	ap-southeast-2	dlf.ap- southeast-2.myhu	HTTPS	

Table 1-1 DLF endpoints

1.4 Constraints

- 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.

IAM user

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

The account name, 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. For more refined access control, create subprojects under a project and create resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

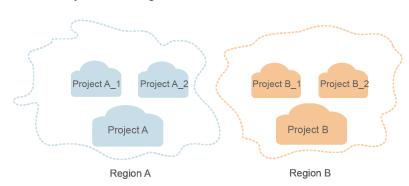


Figure 1-1 Project isolating model

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

Data Development provides self-developed APIs that comply with REST API design specifications. You can fulfill all Data Development functions using those APIs.

3 Calling APIs

3.1 Making an API Request

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

Request URI

A request URI is in the following format:

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

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

Table 3-1 URI parameter description

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints . For example, the endpoint of IAM in the CN-Hong Kong region is iam.ap-southeast-1.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 indicates that a maximum of 10 data records will be displayed.

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

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

Figure 3-1 Example URI



□ NOTE

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

Request Methods

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

Table 3-2 HTTP methods

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

Method	Description
PATCH	Requests the server to update partial content of a specified resource.
	If the resource does not exist, a new resource will be created.

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

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

Request Header

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

Table 3-3 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	e9993fc787d94b6c886cb aa340f9c0f4
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-Coggg1BBIINPXsidG9rZ
X-Language	Request language.	Yes	en_us

□ NOTE

In addition to supporting token-based authentication, public 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.ap-southeast-1.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.

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.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
  "auth": {
     "identity": {
       "methods": [
          "password"
        'password": {
          "user": {
            "name": "username",
            "password": " *******
            "domain": {
               "name": "domainname"
         }
       }
    },
     "scope": {
        "project": {
          }
  }
```

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 the API.

When calling an 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....
```

AK/SK-based Authentication

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

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

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

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or use the signing SDK to sign requests. For details about how to sign requests and use the signing SDK, see API Request Signing Guide.

NOTICE

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

3.3 Response

Status Code

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

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see **Status 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

```
connection → keep-alive

content-type → application/json

date → Tue, 12 Feb 2019 06:52:13 GMT

server → Web Server

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

transfer-encoding → chunked

via → proxy A

x-content-type-options → nosniff

x-download-options → nospen

x-frame-options → SAMEORIGIN

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

| x-subject-token → MIXYXQYJKoZIhvcNAQcCoIIVTjCCGEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w088wGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6ijjhwMTktMDItMTNUMC

ij3Kls67gknpVNRbW2c25eb78SZOkqjACglkqOlwi4JlGzrpd18LGXK5bdfq4lqHCYb8P4NaY0NYejcAgzJVeFIYtLWT1CSCOdzxKZmlQHQlg2HBqHdgjZO9ftEbL5dMhdayi-33wEI

x+IHCE9187o-k-9-

i+CMZ5EB7bUGG5Uj6eRASXI1jipPEGA270g1FruooL6jqglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RZT6MUbpvGw-oPNFYxJECKnoH3HRozvOvN--n5d6Nbxg==

x-xsss-protection → 1; mode=block;
```

(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**.

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

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

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

4 Getting Started

This section describes the procedure for executing a job by calling the API.

- 1. Use the token authentication to obtain the user token, which will be put into the request header for authentication in a subsequent request.
- 2. Execute the job.

Prerequisites

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

Procedure

The following value settings are used as an example:

- IAM endpoint: *iam_endpoint*
- Data Development endpoint: dlf_endpoint
- Project ID: project_id

Perform the following operations to create an application.

1. Use the token authentication API to obtain the user token and set it as an environment variable.

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

The value of **X-Subject-Token** in the response header is the token.

X-Subject-Token:MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...

Run the following command to set the token as an environment variable for future use:

export Token={X-Subject-Token}

Replace **X-Subject-Token** with the token obtained in the preceding step. export X-Auth-Token=MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...

2. Call the API to execute the job.

```
curl -H "Content-Type:application/json" https://{dlf_endpoint}/v1.0/{project_id}/pipelines/run-pipeline -X POST -d ' {
"pipelineId": "b9636fa3-048f-495e-a33f-74b2744801ec"
}
```

If the returned status code is 200, the job is successfully executed.

5 DLF APIS

5.1 Job Development APIs

5.1.1 Running a Job

Function

This API is used to run a job.

URI

POST /v1.0/{project_id}/pipelines/run-pipeline

Table 5-1 lists the parameters.

Table 5-1 Parameter description

Parameter	Mand atory	Туре	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request

Example request:

```
POST https://{Endpoint}:{port}/v1.0/{project_id}/pipelines/run-pipeline {
    "jobId": 54022
}
```

Table 5-2 lists the parameters.

Table 5-2 Parameter description

Parameter	Mandatory	Туре	Description
jobId	Yes	Int	Unique identifier of the job. For details about how to obtain the job ID, see Obtaining a Job ID.

Response

None.

Status Code

- If the operation is successful, status code 200 is returned. For details about status codes, see **Status Codes**.
- If the operation failed, identify the failure cause according to the response body and the instructions in **Error Codes**.

5.1.2 Stopping Scheduling a Job

Function

This API is used to stop scheduling a job.

URI

POST /v1.0/{project_id}/pipelines/stop-pipeline

Table 5-3 lists the parameters.

Table 5-3 Parameter description

Parameter	Mand atory	Туре	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID.

Request

Example request:

```
POST https://{Endpoint}:{port}/v1.0/{project_id}/pipelines/stop-pipeline {
    "jobId": 54022
}
```

Table 5-4 lists the parameters.

Table 5-4 Parameter description

Parameter	Mandatory	Туре	Description
jobId	Yes	Int	Unique identifier of the job. For details about how to obtain the job ID, see Obtaining a Job ID.

Response

None.

Status Code

- If the operation is successful, status code 200 is returned. For details about status codes, see **Status Codes**.
- If the operation failed, identify the failure cause according to the response body and the instructions in **Error Codes**.

6 Data Structure

Table 6-1 describes common request headers.

Table 6-1 Common request header fields

Parameter	Mandator y	Туре	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 <i>hostname:port number</i> . If the port number is not specified, the default port is used. The default port number for HTTPS is 443.
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.

Parameter	Mandator y	Туре	Description
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 6-2 describes common response headers.

Table 6-2 Common response headers

Parameter	Туре	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.

Permissions Policies and Supported Actions

7.1 Introduction

This chapter describes fine-grained permissions management for your Data Lake Factory (DLF). If your HUAWEI CLOUD account does not need individual IAM users, then you may skip over this chapter.

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

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

■ NOTE

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

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

Supported Actions

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

- Permission: A statement in a policy that allows or denies certain operations.
- Actions: Added to a custom policy to control permissions for specific operations.
- Authorization Scope: A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions supporting both IAM and enterprise projects can be assigned to user groups and take effect in both IAM and Enterprise Management. Policies that only contain actions supporting IAM projects can be assigned to user groups and only take effect for IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Project.
- APIs: REST APIs that can be called in a custom policy.

Table 7-1 Actions

Permissions	Actions	Authorizatio n Scope	APIs
Creating Data Connections	dlf:connection :create	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/ connections
Querying a Connection List	dlf:connection :list	 Supported: IAM projects Enterprise projects Not supported: none 	GET /v1/{project_id}/ connections? offset={offset}&limit={limit}&c onnectionName={connectionN ame}
Querying Connection Details	dlf:connection :get	 Supported: IAM projects Enterprise projects Not supported: none 	GET /v1/{project_id}/ connections/ {connection_name}

Permissions	Actions	Authorizatio n Scope	APIs
Editing a Data Connection	dlf:connection :update	 Supported: IAM projects Enterprise projects Not supported: none 	PUT /v1/{project_id}/ connections/ {connection_name}
Deleting a Data Connection	dlf:connection :delete	 Supported: IAM projects Enterprise projects Not supported: none 	DELETE /v1/{project_id}/ connections/ {connection_name}
Exporting a Data Connection	dlf:connection :export	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/ connections/export
Importing a Data Connection	dlf:connection :import	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/ connections/import
Creating a Script	dlf:script:creat e	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/scripts

Permissions	Actions	Authorizatio n Scope	APIs
Querying a Script	dlf:script:get	 Supported: IAM projects Enterprise projects Not supported: none 	GET /v1/{project_id}/scripts/ {script_name}
Querying a Script List	dlf:script:list	 Supported: IAM projects Enterprise projects Not supported: none 	GET /v1/{project_id}/scripts? offset={offset}&limit={limit}&s criptName={scriptName}
Modifying a Script	dlf:script:upda te	 Supported: IAM projects Enterprise projects Not supported: none 	PUT /v1/{project_id}/scripts/ {script_name}
Deleting a Script	dlf:script:delet e	 Supported: IAM projects Enterprise projects Not supported: none 	DELETE /v1/{project_id}/ scripts/{script_name}
Executing a Script	dlf:script:exec ute	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/scripts/ {script_name}/execute

Permissions	Actions	Authorizatio n Scope	APIs
Querying the Execution Result of a Script Instance	dlf:script:get	 Supported: IAM projects Not supported: Enterprise projects 	GET /v1/{project_id}/scripts/ {script_name}/instances/ {instance_id}
Stopping Executing a Script Instance	dlf:script:canc el	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/scripts/ {script_name}/instances/ {instance_id}/stop
Creating a Resource	dlf:resource:cr eate	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/ resources
Querying a Resource	dlf:resource:g et	 Supported: IAM projects Enterprise projects Not supported: none 	GET /v1/{project_id}/ resources/{resource_id}
Querying a Resource List	dlf:resource:lis t	 Supported: IAM projects Not supported: Enterprise projects 	GET /v1/{project_id}/ resources? offset={offset}&limit={limit}&r esourceName={resourceName }

Permissions	Actions	Authorizatio n Scope	APIs
Modifying a Resource	dlf:resource:u pdate	 Supported: IAM projects Enterprise projects Not supported: none 	PUT /v1/{project_id}/ resources/{resource_id}
Deleting a Resource	dlf:resource:d elete	 Supported: IAM projects Enterprise projects Not supported: none 	DELETE /v1/{project_id}/ resources/{resource_id}
Creating a Job	dlf:job:create	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/jobs
Editing a Job	dlf:job:update	 Supported: IAM projects Not supported: Enterprise projects 	PUT /v1/{project_id}/jobs/ {name}
Viewing a Job List	dlf:job:list	 Supported: IAM projects Enterprise projects Not supported: none 	GET /v1/{project_id}/jobs? jobType={jobType}&offset={off set}&limit={limit}&jobName={j obName}

Permissions	Actions	Authorizatio n Scope	APIs
Viewing Job Details	dlf:job:get	 Supported: IAM projects Not supported: Enterprise projects 	GET /v1/{project_id}/jobs/ {name}
Exporting a Job	dlf:job:export	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/jobs/ {name}/export
Batch Exporting Jobs	dlf:job:export	 Supported: IAM projects Not supported: Enterprise projects 	POST /v1/{project_id}/jobs/ batch-export
Viewing a Job File	dlf:job:get	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/jobs/ check-file
Importing a Job	dlf:job:import	 Supported: IAM projects Not supported: Enterprise projects 	POST /v1/{project_id}/jobs/ import

Permissions	Actions	Authorizatio n Scope	APIs
Starting a Job	dlf:job:startJo b	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/jobs/ {name}/start
Stopping a Job	dlf:job:stopJob	 Supported: IAM projects Not supported: Enterprise projects 	POST /v1/{project_id}/jobs/ {name}/stop
Deleting a Job	dlf:job:delete	 Supported: IAM projects Not supported: Enterprise projects 	DELETE /v1/{project_id}/jobs/ {name}
Executing a Job Immediately	dlf:job:testRun	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/jobs/ {name}/run-immediate
Viewing a Job Instance List	dlf:jobInstanc e:list	 Supported: IAM projects Not supported: Enterprise projects 	GET /v1/{project_id}/jobs/ instances/detail? jobName={jobName}&minPla nTime={minPlanTime}&maxPl anTime={maxPlanTime}&limit ={limit}&offset={offset}&statu s={status}

Permissions	Actions	Authorizatio n Scope	APIs
Stopping a Job Instance	dlf:jobInstanc e:stopInstance	 Supported: IAM projects Not supported: Enterprise projects 	POST /v1/{project_id}/jobs/ {jobName}/instances/ {instanceId}/stop
Retry a Job Instance	dlf:jobInstanc e:retryInstanc e	 Supported: IAM projects Enterprise projects Not supported: none 	POST /v1/{project_id}/jobs/ {jobName}/instances/ {instanceId}/restart

8 Appendix

8.1 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:

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

Table 8-1 describes status codes.

Table 8-1 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	ОК	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	MethodNotAllowe d	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	UnprocessableEn- tity	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	InternalServerEr- ror	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.

8.2 Error Codes

```
The error response is in the following format:
```

```
{
"error_code":"DLF.1001",
"error_msg":"The job not found"
}
```

Table 8-2 describes the error codes.

Table 8-2 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.

8.3 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).

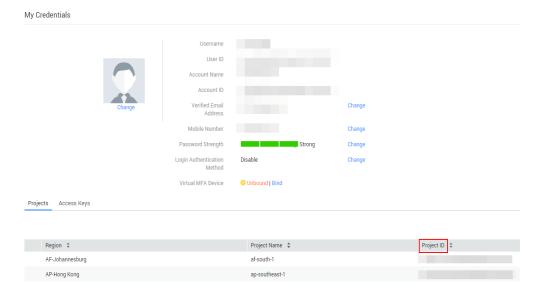
```
"projects": [
  {
     "domain_id": "65382450e8f64ac0870cd180d14e684b",
     "is_domain": false,
     "parent_id": "65382450e8f64ac0870cd180d14e684b",
     "name": "region-name",
     "description": "",
     "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
     },
"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:

- Log in to the management console.
- 2. Hover the mouse over the username and select **Basic Information**.
- On the displayed page, click Manage in Security Credentials.
 On the My Credentials page, view project IDs in the project list.

Figure 8-1 Viewing project IDs



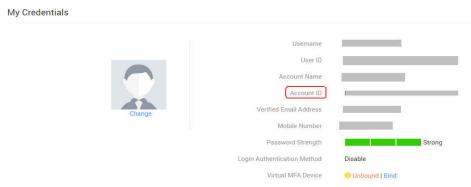
8.4 Obtaining an Account ID

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 8-2 Obtaining an account ID

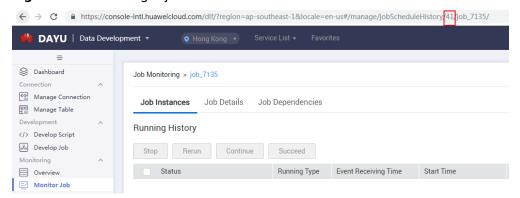


8.5 Obtaining a Job ID

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 8-3 Obtaining a job ID



8.6 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	What's New
2019-04-28	This issue is the first official release.