DataArts Insight

API Reference

Issue 01

Date 2025-05-26

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Before You Start

Welcome to DataArts Insight. DataArts Insight is an all-in-one BI development platform for data visualization that offers visual, real-time, easy-to-use, and secure enterprise intelligence analysis services. It works with various data sources in the cloud and on-premises, offers a wide range of visualization components, and allows for easy data analysis and report building through drag-and-drop layout. It also enables quick customization of exclusive data screens.

You can use the APIs provided in this document to perform various operations on DataArts Insight, including accessing data sources, preparing datasets, analyzing data, and creating large screens. For a full list of supported operations, refer to **Overview**.

Basic Concepts

- Account

  An account is created when a user registers a Huawei ID and enables Huawei Cloud services. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used to perform routine management. For security purposes, create IAM users and grant them permissions for routine management.

- Region

  Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as common regions and dedicated regions. A common region provides common cloud services that should be made available to all tenants. A dedicated region provides services of a specific type or only for specific tenants.

  For details, see **Region and AZ**.

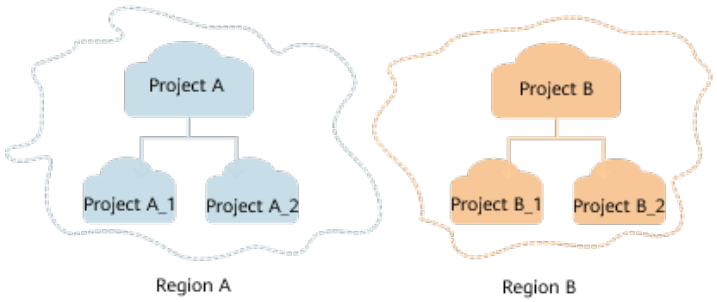
- AZ

  An AZ contains one or more physical data centers. It has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, compute network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to support cross-AZ high-availability systems.

- Project

  Projects group and isolate 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 need more refined access control, create subprojects under a default project and purchase resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

  Figure 1-1 Project isolating model

2 Overview

This section describes the APIs provided by DataArts Insight.

Table 2-1 DataArts Insight APIs

Type	Description
Sharing and Collaboration APIs	These APIs are used to retrieve detailed information about specified resources such as data sources, datasets, dashboards, and large screens, as well as to share and collaborate on these resources.
Workspace APIs	These APIs are used to retrieve information about a workspace, create a workspace, and modify or delete a workspace.
User Tag APIs	These APIs are used to create, edit, and delete user tags, save user tag content, obtain user tag values, and obtain user tag headers.
Dataset Permission APIs	These APIs are used to list dataset permissions, configure dataset permissions, enable or disable dataset permissions, delete permissions, and obtain dataset permission configuration information.
Dataset APIs	These APIs are used to list datasets and directories, obtain dataset details, delete datasets, and save or modify datasets.
Embedded Analytics APIs	These APIs are used to obtain shared embedded analytics information.
Import and Export Parameter APIs	These APIs are used to configure import and export parameters.
Product Instance APIs	These APIs are used to list a user's enabled product instances.
Resource Migration APIs	These APIs are used to export resource packages, obtain import task details, and import resource package files.

Type	Description
Common Resource Query APIs	These APIs are used to list dashboards or large screens.
Data Source APIs	These APIs are used to update, obtain details, delete, list, and create data sources.
Pass Management APIs	These APIs are used to list users, update the user pass status, add users associated with quotas, list users with allocated passes, and delete user allocation data.
Dashboard APIs	These APIs are used to create, update, batch publish, batch take offline, and batch delete dashboards.
Intelligent Analysis Assistant APIs	These APIs are used to list intelligent analysis assistants and ask data questions in natural language.
Large Screen APIs	These APIs are used to list large screen components and obtain the data of specified large screen components.

3 API Calling

3.1 Making an API Request

This section describes the structure of a REST API, and uses IAM's **Obtaining a User Token** API 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 a request header, most programming languages or frameworks require the request URI to be separately transmitted, rather than being conveyed in a request message.

Table 3-1 Parameters in a URI

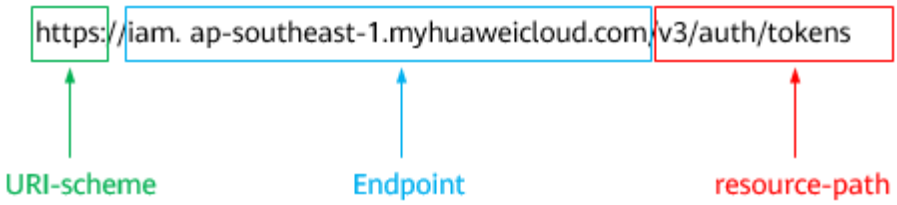
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. Endpoints vary depending on services and regions. For the endpoints of all services, see **Regions and Endpoints**.
resource-path	Access 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 **Obtaining a User Token** API. 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, each API is provided with only a **resource-path** and a request method. This is because the **URI-scheme** value of all APIs is **HTTPS**, and the endpoints in a region are the same. Therefore, the two parts are omitted.

Request Methods

HTTP-based request methods, which are also called operations or actions, specify the type of operations that you are requesting.

Table 3-2 HTTP method

Request Method	Description
GET	Requests a 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 a specific resource, for example, an object.

Request Method	Description
PATCH	Requests a server to update part of specified resources. If the resource is unavailable, the PATCH method is used to create a resource.

Request Header

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

Table 3-3 lists common request header fields.

Table 3-3 Common request headers

Parameter	Description	Mandatory	Example
Host	Specifies the information about the requested server. The value can be obtained from the URL of the service API. The value is **hostname[:port]**. If the port number is not specified, the default port is used. The default port number for **https** is port **443**.	No This header field is mandatory if AK/SK authentication is in use.	code.test.com or code.test.com:443
Content-Type	Specifies the request body type or format. This field is mandatory and its default value is application/json. For other values, the description will be provided for specific APIs.	Yes	application/json
Content-Length	Indicates the length of the request body. The unit is byte.	This field is mandatory for POST and PUT requests, but must be left blank for GET requests.	3495

Parameter	Description	Mandatory	Example
X-Project-ID	Project ID. It is mandatory in multi-project scenarios to obtain tokens for different projects.	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	User token. User token is a response to the API for obtaining a user token (only this API does not require authentication).	This parameter is mandatory only for authentication using tokens.	The following is part of an example token: MIIPAgYJKoZIhvc-NAQcCo...ggg1BBIINPXsidG9rZ
Authorization	Specifies the signature authentication information. The value can be obtained from the request signing result.	This header field is mandatory if AK/SK authentication is in use.	-
X-Sdk-Date	Specifies the time when the request is sent. The time is in *YYYYMMDD*'**T**'*HHMMSS*'**Z**' format. The value is the current Greenwich Mean Time (GMT) of the system.	This header field is mandatory if AK/SK authentication is in use.	20150907T101459Z
X-Language	Request language.	No	en-us

NOTE

In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign 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.

The API for **obtaining a user token** does not require authentication. Therefore, this API only requires adding the Content-Type field. The request with the added Content-Type header is as follows:

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

Request Body

A request body is generally sent in a structured format. It corresponds to **Content-Type** in the request header and transfers data except for the request header.

The request body varies according to the APIs. Certain APIs do not require the request body, such as the GET and DELETE APIs.

For IAM's **Obtaining a User Token** API, obtain the request parameters and parameter description from the API request. The following provides an example request with a body included. Replace ***username***, ***domainname***, ********* (login password), and ***xxxxxxxxxxxxxxxxx*** (project ID) with the actual values. Obtain a project ID from **Regions and Endpoints**.

NOTE

scope specifies where a token takes effect. In the following example, the token takes effect only on the resources specified by the project ID. You can set **scope** to an account or a project under an account. For details, 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": {
                "name": "xxxxxxxxxxxxxxxxx"
            }
        }
    }
}
```

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

3.2 Authentication

API calling is authenticated using either of the following methods:

- Token authentication: Common requests are authenticated using Tokens.
- AK/SK authentication: Requests are encrypted using the access key ID and secret access key (AK/SK pair) to provide higher security. AK/SK authentication is recommended because it provides higher security than token authentication.

Token Authentication

A token specifies certain permissions in a computer system. Authentication using a token adds the token in a request as its header during API calling to obtain permissions to operate APIs through IAM.

> **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.

When calling the **Obtaining a User Token** API, 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 obtaining the token, add the **X-Auth-Token** header in a request to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, add **X-Auth-Token: ABCDEFJ....** in a request as follows:

```
GET https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK Authentication

In authentication using AK/SK, AK/SK is used to sign a request and add the signature in a request as its header for authentication.

> **NOTE**
>
> Authentication using AK/SK supports API requests with a body not larger than 12 MB. For API requests with a larger body, authentication using tokens is recommended.

- AK: access key ID, which is a unique identifier associated with a secret access key and is 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 request senders and prevents their requests from being modified.

In authentication using AK/SK, you can use AK/SK to sign requests based on the signature algorithm or use a dedicated signature SDK to sign the requests.

NOTE

The signature SDK only supports signature, which is different from the SDKs provided by services.

3.3 Returned Values

Status Code

After sending a request, you will receive a response, including the 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 response. For more information, see **Status Codes**.

For example, if status code **201** is returned after calling the **Obtaining a User Token** API, the request is successful.

Response Header

A response header corresponds to a request header, for example, **Content-Type**.

Figure 3-2 shows the request header fields returned after calling the **Obtaining a User Token** API. **x-subject-token** is the desired user token. You can use the token to authenticate other API calls.

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

Response Body

A response body is generally returned in a structured format, corresponding to the **Content-Type** in the response header, and is used to transfer content other than the response header.

The following is part of the response body for the **Obtaining a User Token** API. The following describes only part of the request body.

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "az-01",
.....
```

If an error occurs during API calling, the system returns an error code and a message to you. The following shows the format of an error response body:

```
{

    "error":{
        "code":401,
        "message": "The username or password is wrong.",
        "title": "Unauthorized"
    }
}
```

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

4 Sharing and Collaboration

4.1 Obtaining Resource Attribute Values

Function

This API is used to retrieve detailed information about specified resources such as data sources, datasets, dashboards, and large screens.

URI

GET /v1/{project_id}/authorization/cooperate-authorization/properties

Table 4-1 URI parameter

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

Table 4-2 Query parameters

Parameter	Mandatory	Type	Description
resource_id	Yes	String	Resource ID.

Parameter	Mandatory	Type	Description
resource_type	Yes	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A

Request Parameters

Table 4-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's API. **X-Subject-Token** in the response header is the desired user token. For how to obtain a token, see **Authentication**.

Response Parameters

Status code: 200

Table 4-4 Response body parameters

Parameter	Type	Description
auth_switch_config	Map<String,Boolean>	Switch for permissions. It can be set to **true** or **false** to indicate whether the corresponding switch is turned on. • **edit**: edit permission. • **export**: export permission. • **read**: view permission. • **use**: use permission. **NOTE** • There are dependencies between permissions. Granting **edit** permission automatically associates the **read** and **export** or **use** permission. Granting **export** permission automatically associates the **read** permission. • For dashboards and large screens, there are three permissions: **edit**, **export**, and **read**. For example, **{"edit":false, "read":true, "export":true}** means that **view** and **export** permissions are enabled. • For data sources and datasets, there are two permissions: **edit** and **use**, corresponding to edit and use permissions. For example, **{"use":true, "edit":false}** means that only the **use** permission is enabled.
create_date	Long	Resource creation time.
create_user	String	ID of the user who creates the resource.
create_user_name	String	Name of the resource creator.
id	String	ID of the resource collaboration authorization attribute table.
owner	String	Resource owner ID.
project_id	String	Project ID.
resource_id	String	Resource ID.

Parameter	Type	Description
resource_type	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
update_date	Long	Modification time.
update_user	String	Modifier ID.
update_user_name	String	Modifier name.
workspace_id	String	Workspace ID.

Example Request

Obtains resource attribute values.

```
GET https://{endpoint}/v1/9c3043axxxac4055888643b331a0bxxx/authorization/cooperate-authorization/properties?resource_type=dataset&resource_id=xxxx406b-8662-4adb-a028-a55cxxxx1166
```

Example Response

Status code: 200

OK.

```
{
  "create_date" : 1682576908274,
  "create_user" : "aoeusth",
  "create_user_name" : "xxxx",
  "update_date" : 1683343425273,
```



```
  "update_user" : "aeousth",
  "update_user_name" : "xxxx",
  "id" : "xxxxx1f0-dea2-472c-aa9c-25928f74xxxx",
  "resource_type" : "dataset",
  "resource_id" : "xxxx406b-8662-4adb-a028-a55cxxxx1166",
  "auth_switch_config" : {
    "edit" : false,
    "read" : true
  },
  "owner" : "xxxxx4a863a94b0ba2887b80dcxxxx3",
  "project_id" : "9c3043axxxac4055888643b331a0bxxx",
  "workspace_id" : "66a2c96809ad4f62ba09e6b6eeaa2146"
}
```


Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

4.2 Saving or Modifying Resource Attribute Values

Function

This API is used to save or modify resource attribute values.

URI

POST /v1/{project_id}/authorization/cooperate-authorization/properties

Table 4-5 URI parameter

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

Request Parameters

Table 4-6 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's API. **X-Subject-Token** in the response header is the desired user token. For how to obtain a token, see **Authentication**.

Table 4-7 Request body parameters

Parameter	Mandatory	Type	Description
auth_switch_config	No	Map<String,Boolean>	Switch for permissions. It can be set to **true** or **false** to indicate whether the corresponding switch is turned on. The default value is **false**. • **edit**: edit permission. • **export**: export permission. • **read**: view permission. • **use**: use permission. **NOTE** • There are dependencies between permissions. Granting **edit** permission automatically associates the **read** and **export** or **use** permission. Granting **export** permission automatically associates the **read** permission. • For dashboards and large screens, there are three permissions: **edit**, **export**, and **read**. For example, **{"edit":false, "read":true, "export":true}** means that **view** and **export** permissions are enabled. • For data sources, intelligent analysis assistants, and datasets, there are two permissions: **edit** and **use**, corresponding to edit and use permissions. For example, **{"use":true, "edit":false}** means that only the **use** permission is enabled.
resource_id	Yes	String	Resource ID.

Parameter	Mandatory	Type	Description
resource_type	Yes	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A

Response Parameters

Status code: 200

Table 4-8 Response body parameters

Parameter	Type	Description
auth_switch_config	Map<String,Boolean>	Switch for permissions, that is, the sharing and collaboration configuration item. It can be set to **true** or **false** to indicate whether the corresponding switch is turned on. • **edit**: edit permission. • **export**: export permission. • **read**: view permission. • **use**: use permission. **NOTE** • There are dependencies between permissions. Granting **edit** permission automatically associates the **read** and **export** or **use** permission. Granting **export** permission automatically associates the **read** permission. • For dashboards and large screens, there are three permissions: **edit**, **export**, and **read**. For example, **{"edit":false, "read":true, "export":true}** means that **view** and **export** permissions are enabled. • For data sources, datasets, and domains, there are two permissions: **edit** and **use**, corresponding to edit and use permissions. For example, **{"use":true, "edit":false}** means that only the **use** permission is enabled.
create_date	Long	Creation time.
create_user	String	Creator ID.
create_user_name	String	Creator name.
id	String	ID of the resource collaboration authorization attribute table.
owner	String	Resource owner ID.
project_id	String	Project ID.
resource_id	String	Resource ID.

Parameter	Type	Description
resource_type	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
update_date	Long	Modification time
update_user	String	Modifier ID.
update_user_name	String	Modifier name.
workspace_id	String	Workspace ID.

Example Request

Save or modify resource sharing and collaboration attribute values.

```
POST https://{endpoint}/v1/xxxxx3ab4dac4055888643b331xxxxx/authorization/cooperate-authorization/properties
{
  "auth_switch_config" : "{"read": true, "edit": true, "export": true}",
  "resource_id" : "d0d7919c-xxxx-462e-xxxx-da82bf8be44e",
  "resource_type" : "dashboard"
}
```

Example Response

Status code: 200

OK.


```
{
  "auth_switch_config" : "{"read": true, "edit": true, "export": true}",
  "create_date" : 1688022840899,
  "create_user" : "de9cb4xxxxa94b0ba2887b80xxxxxxxx",
  "create_user_name" : "***_01",
  "id" : "78215911-xxxx-478b-xxxx-9cb7f0407598",
  "owner" : "dxxxxa863a94xxxa2887b80xxxxxxxx",
  "project_id" : "xxx3ab4dac40558886xxxxx1a0b227",
  "resource_id" : "d0xxx19c-4e87-46xx-bxx2-da82bfxxxx4e",
  "resource_type" : "dashboard",
  "update_date" : 1688023886860,
  "update_user" : "de9cb4a863a94b0ba2887b80xxxxxxxx",
  "update_user_name" : "ei_dlv_***_01",
  "workspace_id" : "6xxxx96809adxxxxxa09e6bxxxx2146"
}
```


Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

4.3 Listing Authorization Records

Function

This API is used to list authorization records.

URI

GET /v1/{project_id}/authorization/cooperate-authorization/rules

Table 4-9 URI parameter

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

Table 4-10 Query parameters

Parameter	Mandatory	Type	Description
auth_level	No	String	Level of the authorization entity. • **user**: user. • **group**: user group.
filter_authed	No	Boolean	Whether to filter authorized entities. The default value is **true**, indicating that only authorized users and user groups are displayed. Reserved field.
resource_id	Yes	String	Resource ID
resource_type	Yes	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
auth_name	No	String	Authorization entity name.
limit	No	Integer	Number of records displayed on each page.

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset.
sort_dir	No	String	**asc**: ascending order; **desc**: descending order.

Request Parameters

Table 4-11 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's API. **X-Subject-Token** in the response header is the desired user token. For how to obtain a token, see **Authentication**.

Response Parameters

Status code: 200

Table 4-12 Response body parameters

Parameter	Type	Description
page_data	Array of objects	Pagination list content. For parameter types, see **CooperateAuthorizationRuleVO**.
count	Integer	Total number of records.

Table 4-13 CooperateAuthorizationRuleVO

Parameter	Type	Description
auth_id	String	Authorization entity ID.
auth_level	String	Authorization entity level. The options include: • **user**: user. • **group**: user group.

Parameter	Type	Description
auth_name	String	Authorization entity name.
authed	Boolean	Whether it is authorized.
authority	String	Permissions. The options include: • **edit,export**: edit and export permissions. • **edit**: edit permission. • **export**: export permission. • **read**: view permission. • **use**: use permission. **NOTE** Permissions depend on each other. • Granting the **edit** permission will automatically grant the **use** and **read** permissions. • Granting the **export** permission will automatically grant the **use** and **read** permissions. • The **export** and **edit** operations do not depend on each other.
create_date	Long	Creation time.
create_user	String	Creator ID.
create_user_name	String	Creator name.
id	String	ID of an authorization record, which has no actual meaning.
is_owner	Boolean	Whether the user is the resource owner.
resource_id	String	Resource ID.

Parameter	Type	Description
resource_type	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
sort	Integer	Sorting field, which has no actual meaning.
update_date	Long	Modification time.
update_user	String	Modifier ID.
update_user_name	String	Modifier name.

Example Request

List authorization records.

```
GET https://{endpoint}/v1/xxxx3ab4dac4055888643b331axxxx/authorization/cooperate-authorization/rules?resource_type=dataset&resource_id=f8e5406b-8662-4adb-a028-a55c76ae1166&auth_level=user&filer_authed=false&limit=20&offset=0&sort_dir=asc
```

Example Response

Status code: 200

OK.

```
{
  "page_data" : [ {
    "create_date" : 1690593374608,
    "create_user" : "xxxxxf45cea249a2aa648a20e46xxxxx",
```



```
  "create_user_name" : "xxxxxx",
  "update_date" : 1690593374608,
  "update_user" : "558def45cea249a2aa648a20e46xxxxx",
  "update_user_name" : "xxxxx",
  "id" : "73281a3b-a828-494a-bb0d-e88c688a5e81",
  "resource_type" : "dashboard",
  "resource_id" : "e3158b30-30bc-495a-b0d8-59e66e1xxxxx",
  "auth_level" : "user",
  "auth_id" : "xxxxf45cea249a2aa648a20e46xxxx",
  "auth_name" : "xxxxxx",
  "is_owner" : true,
  "authed" : true,
  "sort" : 4,
  "authority" : "edit"
 }, {
  "create_date" : 1690593374608,
  "create_user" : "xxxxxf45cea249a2aa648a20e46xxxxx",
  "create_user_name" : "xxxxxx",
  "update_date" : 1690593374608,
  "update_user" : "558def45cea249a2aa648a20e46xxxxx",
  "update_user_name" : "xxxxx",
  "id" : "73281a3b-a828-494a-bb0d-e88c688a5e81",
  "resource_type" : "dashboard",
  "resource_id" : "e3158b30-30bc-495a-b0d8-59e66e1xxxxx",
  "auth_level" : "user",
  "auth_id" : "xxxxf45cea249a2aa648a20e46xxxx",
  "auth_name" : "xxxxxx",
  "is_owner" : true,
  "authed" : true,
  "sort" : 4,
  "authority" : "edit,export"
 } ],
 "count" : 2147483647
}
```


Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

4.4 Batch Saving, Modifying, and Deleting Authorization Records

Function

This API is used to save, modify, and delete specified self-developed authorization rule records in batches.

URI

POST /v1/{project_id}/authorization/cooperate-authorization/rules/batch-save

Table 4-14 URI parameter

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

Request Parameters

Table 4-15 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's API. **X-Subject-Token** in the response header is the desired user token. For how to obtain a token, see **Authentication**.

Table 4-16 Request body parameters

Parameter	Mandatory	Type	Description
[Array element]	Yes	Array of **CooperateAuthorizationRuleDto** objects	Authorization rules.

Table 4-17 CooperateAuthorizationRuleDto

Parameter	Mandatory	Type	Description
auth_id	No	String	User or user group ID of the authorization entity.
auth_level	No	String	Authorization entity level. The options include: • **user**: user. • **group**: user group.

Parameter	Mandatory	Type	Description
authority	No	String	Permissions. The options include: • **edit,export**: edit and export permissions. • **edit**: edit permission. • **export**: export permission. • **read**: view permission. • **use**: use permission. **NOTE** Permissions depend on each other. • Granting the **edit** permission will automatically grant the **use** and **read** permissions. • Granting the **export** permission will automatically grant the **use** and **read** permissions. • The **export** and **edit** operations do not depend on each other.
resource_id	No	String	Resource ID.

Parameter	Mandatory	Type	Description
resource_type	No	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A

Response Parameters

Status code: 200

Table 4-18 Response body parameter

Parameter	Type	Description
message	String	Message content of the general Message structure.

Example Request

Save, modify, and delete specified self-developed authorization rule records in batches.

```
POST https://{endpoint}/v1/xxxx3ab4dac4055888643b331axxxx/authorization/cooperate-authorization/rules/batch-save
```



```
[ {
  "auth_id" : "de9cb4xxxx4b0ba2887b80dc18d273",
  "auth_level" : "user",
  "authority" : "edit",
  "resource_id" : "d0d7919c-xxxx-462e-xxxx-da82bf8be44e",
  "resource_type" : "dashboard"
}, {
  "auth_id" : "de9cb4axxxxa2887b80dc18d273",
  "auth_level" : "user",
  "authority" : "edit,export",
  "resource_id" : "d0d7919c-4e87-xxxx-xxxx-da82bf8be44e",
  "resource_type" : "dashboard"
} ]
```

Example Response

The request is successful, and **success** is returned.

```
{
  "message" : "success"
}
```

Status Codes

Status Code	Description
200	The request is successful, and **success** is returned.

Error Codes

See **Error Codes**.

5 Workspace

5.1 Querying Workspaces

Function

This API is used to query workspaces.

You can only view workspaces that are accessible within your authorized permissions.

URI

GET /v1/{project_id}/instances/{instance_id}/workspaces

Table 5-1 URI parameter

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.
project_id	Yes	String	Project ID. For how to obtain the project ID, see **Obtaining a Project ID**.

Table 5-2 Query parameters

Parameter	Mandatory	Type	Description
name	No	String	Workspace name. Fuzzy search is supported. Value range: [0, 100]

Parameter	Mandatory	Type	Description
offset	No	Integer	Start offset of returned entries. Value range: [0, 2147483647] Default value: **0**
limit	No	Integer	Number of returned entries. If not transferred, the first 10 records are queried by default. Value range: [0, 2147483647] Default value: **10**

Request Parameters

Table 5-3 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's API. **X-Subject-Token** in the response header is the desired user token. For how to obtain a token, see **Authentication**.

Response Parameters

Status code: 200

Table 5-4 Response body parameters

Parameter	Type	Description
count	Integer	Total number of entries.
page_data	Array of objects	Data on the current page. The element type is **WorkspaceVO**. For parameter types, see **WorkspaceVO**.

Table 5-5 WorkspaceVO

Parameter	Type	Description
configs	Map<String,String>	configs.
create_time	Long	Workspace creation time.
create_user	String	Workspace creator.
description	String	Workspace description.
domain_id	String	Domain ID.
eps_id	String	ID of the enterprise project the workspace belongs to.
id	String	Workspace ID.
instance_id	String	Instance ID.
is_default	Integer	Whether to set the workspace as the default one. The value can be **0** or **1**.
name	String	Workspace name.
owner_name	String	Workspace owner name.
project_id	String	ID of the project the workspace belongs to.
update_time	Long	Workspace update time.
update_user	String	User who updates the workspace.

Example Request

Query workspaces under a project.

```
/v1/{project_id}/instances/{instance_id}/workspaces?offset=0&limit=10name=
```

Example Response

```
{
  "count" : 17,
  "page_data" : [ {
    "configs" : {
                "mode": "0",
                "worksAuthorized": "0",
                "worksPublic": "1",
                "onlyAdminCreateDatasource": "0",
                "isPin": "1",
                "fieldShowType": "0",
                "worksView": "0"
            },
    "create_time" : 1676984980510,
    "create_user" : "xxxxxx",
    "description" : "Description workspace information",
    "domain_id" : "xxxxxxx",
```



```
  "eps_id" : "xxxxxxxxxx",
  "id" : "xxxxxxxxxxxx",
  "instance_id" : "xxxx7d17c41c414dabaa08f47c7dxxxx",
  "is_default" : 1,
  "name" : "Example workspace name",
  "owner_name" : "xxxxxx",
  "project_id" : "xxxxxxxb4dac4055888643b3xxxxxx",
  "update_time" : 1687167926377,
  "update_user" : "xxxxxxxx"
 } ]
}
```

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.2 Creating a Workspace

Function

This API is used to create a workspace.

URI

POST /v1/{project_id}/instances/{instance_id}/workspaces

Table 5-6 URI parameters

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Request Parameters

Table 5-7 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's API. **X-Subject-Token** in the response header is the desired user token. For how to obtain a token, see **Authentication**.

Table 5-8 Request body parameters

Parameter	Mandatory	Type	Description
configs	No	Map<String,String>	Workspace attribute configuration. • If **default_dataset_permission** is set to **1**, all dataset data can be viewed when row-level permissions are not enabled. When set to **0**, no dataset data can be viewed when row-level permissions are not enabled.
name	Yes	String	Workspace name.
description	No	String	Workspace description.
eps_id	Yes	String	ID of the enterprise project the workspace belongs to.

Response Parameters

Status code: 200

Table 5-9 Response body parameters

Parameter	Type	Description
configs	Map<String,String>	Workspace attribute configuration. • If **default_dataset_permission** is set to **1**, all dataset data can be viewed when row-level permissions are not enabled. When set to **0**, no dataset data can be viewed when row-level permissions are not enabled.
create_time	Long	Creation time.
create_user	String	Creator.
description	String	Workspace description.
domain_id	String	ID of the account the workspace belongs to.
eps_id	String	ID of the enterprise project the workspace belongs to.
id	String	Workspace ID.
instance_id	String	Instance ID.
is_default	Integer	Whether the workspace is the default one. The value can be **0** or **1**.
name	String	Workspace name.
owner_name	String	Workspace owner name.
project_id	String	ID of the project the workspace belongs to.
update_time	Long	Time when workspace details were updated.
update_user	String	Owner who has updated workspace details.

Example Request

Create a workspace under a project.

```
/v1/{project_id}/instances/{instance_id}/workspaces
{
  "configs" : {
    "mode" : "0",
    "field_show_type" : "0",
    "only_admin_create_datasource" : "0",
    "works_authorized" : "0",
    "works_public" : "1",
    "works_view" : "0"
  },
  "name": "Test name",
  "description": "This is a test description.",
  "eps_id" : "0"
}
```


Example Response


```
{
  "configs" : {
     "mode" : "0",
     "field_show_type" : "0",
     "only_admin_create_datasource" : "0",
     "works_authorized" : "0",
     "works_public" : "1",
     "works_view" : "0"
   }
,
  "create_time" : 1676984980510,
  "create_user" : "xxxxxx",
  "description" : "Description workspace information",
  "domain_id" : "xxxxxxx",
  "eps_id" : "xxxxxxxxxx",
  "id" : "xxxxxxxxxxxx",
  "instance_id" : "xxxx7d17c41c414dabaa08f47c7dxxxx",
  "is_default" : 1,
  "name" : "Example workspace name",
  "owner_name" : "xxxxxx",
  "project_id" : "xxxxxxxb4dac4055888643b3xxxxxx",
  "update_time" : 1687167926377,
  "update_user" : "xxxxxxxx"
}
```


Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.3 Modifying a Workspace

Function

This API is used to modify workspaces.

URI

PUT /v1/{project_id}/instances/{instance_id}/workspaces/{workspace_id}

Table 5-10 URI parameters

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.

Parameter	Mandatory	Type	Description
workspace_id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Request Parameters

Table 5-11 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's API. **X-Subject-Token** in the response header is the desired user token. For how to obtain a token, see **Authentication**.

Table 5-12 Request body parameters

Parameter	Mandatory	Type	Description
configs	No	Map<String,String>	Workspace attribute configuration. • If **default_dataset_permission** is set to **1**, all dataset data can be viewed when row-level permissions are not enabled. When set to **0**, no dataset data can be viewed when row-level permissions are not enabled.
name	Yes	String	Workspace name.
description	No	String	Workspace description.
eps_id	Yes	String	ID of the enterprise project the workspace belongs to.

Response Parameters

Status code: 200

Table 5-13 Response body parameter

Parameter	Type	Description
id	String	Workspace ID.

Example Request

Modify the information of the workspace with ID **xxxxaaa59ce841f8b4cc2c1305c1xxxx** in a project.

```
/v1/{project_id}/instances/{instance_id}/workspaces/xxxxaaa59ce841f8b4cc2c1305c1xxxx

{
  "configs" : {
    "mode" : "0",
    "field_show_type" : "0",
    "only_admin_create_datasource" : "0",
    "works_authorized" : "0",
    "works_public" : "1",
    "works_view" : "0"
  },
  "name": "Test name",
  "description": "This is a test description.",
  "eps_id" : "0"
}
```

Example Response

```
{
  "id" : "xxxxaaa59ce841f8b4cc2c1305c1xxxx"
}
```

Status Codes

Status Code	Description
200	-

Error Codes

See **Error Codes**.

5.4 Deleting a Workspace

Function

This API is used to delete workspaces.

URI

DELETE /v1/{project_id}/instances/{instance_id}/workspaces/{workspace_id}

Table 5-14 URI parameters

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.
workspace_id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
project_id	Yes	String	Project ID. For how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 5-15 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's API. **X-Subject-Token** in the response header is the desired user token. For how to obtain a token, see **Authentication**.

Response Parameters

Status code: 200

Table 5-16 Response body parameter

Parameter	Type	Description
message	String	Message content of the general Message structure.

Example Request

Delete a workspace under a project.

```
/v1/{project_id}/instances/{instance_id}/workspaces/{workspace_id}
```


Example Response

```
{
    "status_code": 200,
    "message": null,
    "is_success": true
}
```

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

6 User Tags

6.1 Creating a User Tag

Function

This API is used to create user tags.

URI

POST /v1/{project_id}/tags

Table 6-1 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	project_id

Request Parameters

Table 6-2 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 6-3 Request body parameters

Parameter	Mandatory	Type	Description
description	No	String	description
name	Yes	String	name
type	Yes	String	**Explanation** Tag type, which can be user tag or user group tag. **Constraints** N/A **Value range** The enumerated values are: • **user**: user tag. • **userGroup**: user group tag. **Default value** N/A
default_value_type	Yes	String	Default tag value type. The default value is **NULL**. • **NULL**: Null. • **ALL**: All values. • **ENUM**: Enumerated values. **default_value** must be set to an enumerated value.

Response Parameters

Status code: 200

Table 6-4 Response body parameter

Parameter	Type	Description
{Custom key}	Map<String,String>	

Example Request

Create a user tag.

```
POST https://{endpoint}/v1/{project_id}/tags

{
  "name": "Test 20241127",
  "description": "Added for test",
  "type" : "user",
```



```
  "default_value_type" : "ENUM",
  "default_value": ["Test"]
}
```

Example Response

None

Status Codes

Status Code	Description
200	The user tag is successfully created.

Error Codes

See **Error Codes**.

6.2 Editing a User Tag

Function

This API is used to edit user tags.

URI

PUT /v1/{project_id}/tags/{tag_id}

Table 6-5 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	project_id
tag_id	Yes	String	tag_id

Request Parameters

Table 6-6 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 6-7 Request body parameters

Parameter	Mandatory	Type	Description
description	No	String	description
name	Yes	String	name
type	Yes	String	**Explanation** Tag type, which can be user tag or user group tag. **Constraints** N/A **Value range** The enumerated values are: • **user**: user tag. • **userGroup**: user group tag. **Default value** N/A
default_value_type	Yes	String	Default tag value type. The default value is **NULL**. • **NULL**: Null. • **ALL**: All values. • **ENUM**: Enumerated values. **default_value** must be set to an enumerated value.

Response Parameters

Status code: 200

Table 6-8 Response body parameter

Parameter	Type	Description
data	Boolean	Whether the request is successful. The options are **true** and **false**.

Example Request

```
PUT https://{endpoint}/v1/{project_id}/tags/{tag_id}

{
  "name": "Test 20241127",
  "description": "Added for test",
  "type" : "user",
  "default_value_type" : "ENUM",
  "default_value": ["Test"]
}
```

Example Response

None

Status Codes

Status Code	Description
200	The user tag is successfully edited.

Error Codes

See **Error Codes**.

6.3 Deleting a User Tag

Function

This API is used to delete user tags.

URI

DELETE /v1/{project_id}/tags/{tag_id}

Table 6-9 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	project_id
tag_id	Yes	String	tag_id

Request Parameters

Table 6-10 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 6-11 Response body parameter

Parameter	Type	Description
data	Boolean	Whether the request is successful. The options are **true** and **false**.

Example Request

Delete a user tag.

```
DELETE https://{{host}}/v1/{project_id}/tags/{tag_id}
```

Example Response

None

Status Codes

Status Code	Description
200	The user tag is successfully deleted.

Error Codes

See **Error Codes**.

6.4 Saving User Tag Content (by User)

Function

This API is used to save user tag content by user.

URI

PUT /v1/{project_id}/tags/{tag_id}/values

Table 6-12 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	project_id
tag_id	Yes	String	tag_id

Request Parameters

Table 6-13 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 6-14 Request body parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	userId
value	No	Array of strings	value

Parameter	Mandatory	Type	Description
value_type	No	String	Default tag value type. The default value is **NULL**. • **DEFAULT**: Default value, which is the default tag value. • **NULL**: Null. • **ALL**: All values. • **ENUM**: Enumerated values. **default_value** must be set to an enumerated value.

Response Parameters

Status code: 200

Table 6-15 Response body parameter

Parameter	Type	Description
{Custom key}	Map<String,String>	

Example Request

The user tag content is saved.

```
PUT https://{endpoint}/v1/{project_id}/tags/{tag_id}/values

{
  "user_id" : "bf32ac8de8c64d738caceede9a00d136",
  "value_type" : "ENUM",
  "value" : [ "10086" ]
}
```

Example Response

Status code: 200

The user tag content is successfully saved.

```
{
  "data" : true
}
```


Status Codes

Status Code	Description
200	The user tag content is successfully saved.

Error Codes

See **Error Codes**.

6.5 Obtaining User Tag Values

Function

This API is used to obtain user tag values.

URI

GET /v1/{project_id}/tags/value

Table 6-16 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	project_id

Table 6-17 Query parameters

Parameter	Mandatory	Type	Description
type	Yes	String	**Explanation** Tag type, which can be user tag or user group tag. **Constraints** N/A **Value range** The enumerated values are: • **user**: user tag. • **userGroup**: user group tag. **Default value** N/A
offset	No	Integer	Start offset of returned items. The default value is **0**.

Parameter	Mandatory	Type	Description
limit	No	Integer	Number of returned records. The default value is **10**.

Request Parameters

Table 6-18 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 6-19 Response body parameters

Parameter	Type	Description
count	Integer	Total data volume.
page_data	Array of **UserTagValueItemVO** objects	Data.

Table 6-20 UserTagValueItemVO

Parameter	Type	Description
user_id	String	User ID.
account_name	String	Username.

Parameter	Type	Description
type	String	**Explanation** Tag type, which can be user tag or user group tag. **Constraints** N/A **Value range** The enumerated values are: • **user**: user tag. • **userGroup**: user group tag. **Default value** N/A
tag_values	Map<String,Array <String>>	User tag value.
tag_value_info	Map<String,**UserTagValueVo**>	Tag value information.

Table 6-21 UserTagValueVo

Parameter	Type	Description
user_id	String	User or user group ID.
account_name	String	Name of a user or user group.
type	String	**Explanation** Tag type, which can be user tag or user group tag. **Constraints** N/A **Value range** The enumerated values are: • **user**: user tag. • **userGroup**: user group tag. **Default value** N/A
from_type	String	Source type. The options are: • **IAM**: IAM user or user group. • **CUSTOM**: Custom user group. • **SELF-BUILT**: custom user.
group_path	Array of strings	User group-level path.

Parameter	Type	Description
create_date	Integer	Creation timestamp.
update_date	Integer	Update timestamp.
value_type	String	Default tag value type. The default value is **NULL**. • **DEFAULT**: Default value, which is the default tag value. • **NULL**: Null. • **ALL**: All values. • **ENUM**: Enumerated values. **default_value** must be set to an enumerated value.
value	Array of strings	Enumerated value.

Example Request

Obtain the values of user tags.

```
GET https://{endpoint}/v1/{project_id}/tags?type=user
```

Example Response

None

Status Codes

Status Code	Description
200	User tags are successfully listed.

Error Codes

See **Error Codes**.

6.6 Obtaining a User Tag Header

Function

This API is used to obtain user tag headers.

URI

GET /v1/{project_id}/tags/head

Table 6-22 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	project_id

Table 6-23 Query parameters

Parameter	Mandatory	Type	Description
type	Yes	String	**Explanation** Tag type, which can be user tag or user group tag. **Constraints** N/A **Value range** The enumerated values are: • **user**: user tag. • **userGroup**: user group tag. **Default value** N/A
offset	No	Integer	Start offset of returned items. The default value is **0**.
limit	No	Integer	Number of returned records. The default value is **10**.

Request Parameters

Table 6-24 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 6-25 Response body parameters

Parameter	Type	Description
count	Integer	Number of data records.
page_data	Array of **UserTagVo** objects	Data.

Table 6-26 UserTagVo

Parameter	Type	Description
tag_description	String	tagDescription
tag_id	String	tagId
tag_name	String	tagName
tag_type	String	**Explanation** Tag type, which can be user tag or user group tag. **Constraints** N/A **Value range** The enumerated values are: • **user**: user tag. • **userGroup**: user group tag. **Default value** N/A

Example Request

Obtain user tag headers.

```
GET https://{endpoint}/v1/{project_id}/tags?type=user
```

Example Response

None

Status Codes

Status Code	Description
200	User tags are successfully listed.

Error Codes

See **Error Codes**.

6.7 Exporting User Tags

Function

This API is used to export user tags to an Excel file for batch editing.

URI

GET /v1/{project_id}/tags/download

Table 6-27 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Explanation** Project ID. For how to obtain the project ID, see **Obtaining a Project ID**. **Constraints** N/A **Value range** N/A **Default value** N/A

Table 6-28 Query parameters

Parameter	Mandatory	Type	Description
type	Yes	String	**Explanation** User type. **Constraints** N/A **Value range** • **user**: user. • **userGroup**: user group. **Default value** N/A

Request Parameters

Table 6-29 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	**Explanation** User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token. **Constraints** N/A **Value range** N/A **Default value** N/A

Response Parameters

Status code: 200

Table 6-30 Response body parameter

Parameter	Type	Description
-	File	**Explanation** The tags of the users or user groups under the tenant have been successfully exported to an Excel file. **Constraints** N/A **Value range** N/A **Default value** N/A

Example Request

Export user tags to an Excel file.

```
https://{endpoint}/v1/{project_id}/tags/download?type=user
```


Example Response

None

Status Codes

Status Code	Description
200	The tags of the users or user groups under the tenant have been successfully exported to an Excel file.

Error Codes

See **Error Codes**.

6.8 Importing User Tags

Function

This API is used to modify user or user group tags in batches.

NOTE

- The import file must be an Excel file.
- The number of entries per batch import is limited, with a default limit of 1,000 entries.
- The number of new tags is validated according to the logic of the new tag API. If the number of tags exceeds the limit, the import will fail.
- The validation rules for tag names and tag values are consistent with those for adding new tags or tag values.
- The **account_id** field must be present during import. The value is carried from the export file and serves as the unique ID for the user or user group. It cannot be edited or modified.

URI

POST /v1/{project_id}/tags/import

Table 6-31 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Explanation** Project ID. For how to obtain the project ID, see **Obtaining a Project ID**. **Constraints** N/A **Value range** N/A **Default value** N/A

Request Parameters

Table 6-32 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	**Explanation** User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token. **Constraints** N/A **Value range** N/A **Default value** N/A

Table 6-33 FormData parameters

Parameter	Mandatory	Type	Description
file	Yes	File	**Explanation** File that contains the user or user group tags to be imported, which must be in Excel format. **Constraints** • Must be an Excel file. • The file size must not exceed 10 MB. • The number of rows for users or user groups to be processed during import must not exceed 1,000. **Value range** N/A **Default value** N/A
type	Yes	String	**Explanation** User type. **Constraints** N/A **Value range** • **user**: user. • **userGroup**: user group. **Default value** N/A

Response Parameters

Status code: 200

Table 6-34 Response body parameter

Parameter	Type	Description
data	Boolean	**Explanation** Execution result. **Constraints** N/A **Value range** • **true**: success. • **false**: failure. **Default value** N/A

Example Request

None

Example Response

Status code: 200

The tags are successfully imported.

```
{
  "data" : true
}
```

Status Codes

Status Code	Description
200	The tags are successfully imported.

Error Codes

See **Error Codes**.

7 Dataset Permissions

7.1 Listing Dataset Permissions

Function

This API is used to list dataset permissions.

URI

GET /v1/{project_id}/datasets/{dataset_id}/permissions

Table 7-1 URI parameter

Parameter	Mandatory	Type	Description
dataset_id	Yes	String	Dataset ID.
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Table 7-2 Query parameters

Parameter	Mandatory	Type	Description
permission_type	Yes	String	Permission type. The options are **ROW** (row permission) and **COLUMN** (column permission).

Parameter	Mandatory	Type	Description
offset	Yes	Integer	Offset of pagination. The value **0** indicates no offset, that is, the first page. The value **1** indicates that the offset is 1, that is, the second page.
limit	Yes	Integer	Maximum number of returned items allowed.
sort_key	No	String	Sorting field. The value is **isOpen**, indicating that whether to enable row and column permissions.
sort_dir	No	String	Sorting type. The options are **desc** and **asc**. **desc** indicates the descending order, and **asc** indicates the ascending order.

Request Parameters

Table 7-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Table 7-4 Response body parameters

Parameter	Type	Description
count	Integer	Total data volume.
page_data	Array of objects	Data. For details, see **Table 7-5**.

Table 7-5 page_data

Parameter	Type	Description
id	String	Primary key ID.
name	String	Name.
dataset_id	String	Dataset ID.
is_open	Boolean	Whether to enable row and column permissions. The default value is **true**.
permission_type	String	Permission type. **ROW** indicates that permissions are restricted by row in a database table, and **COLUMN** indicates that permissions are restricted by column in a database table.
rule_type	String	Rule type. When **permission_type** is set to **ROW**, the options are **BY_CONDITION** and **BY_TAG**. When **permission_type** is set to **COLUMN**, the options are **FORBID** and **MASK**.
rule_scope	String	Rule range. **ALL** indicates that the rule applies to all users. **ALL_NO** indicates that the rule does not apply to any user. **SPECIFIED** indicates that the rule only applies to specified users. **SPECIFIED_NOT** indicates that the rule does not apply to specified users.
rule_user	object	Description of the user the rule applies to. For details, see **Table 7-6**.
rule_content	object	Detailed information about row and column permissions. For row permissions, the **DatasetRowPermissionRuleVo** model is used. For column permissions, the **DatasetColumnPermissionRuleVo** model is used. For details, see **Table 7-7**.
display_fields	object	Display fields. For details, see **Table 7-11**.

Table 7-6 rule_user

Parameter	Type	Description
users	Array of strings	User.
user_groups	Array of strings	User group.

Table 7-7 rule_content

Parameter	Type	Description
last	Integer	End position of characters to be processed under custom circumstances.
first	Integer	Start position of characters to be processed under custom circumstances.
mask_type	String	Data masking type.
column_ids	Array of strings	List of effective field IDs.
special_words	Array of strings	Special character list specified by the data masking type.
logic_operator	String	Relationship between sub-conditions at the current level. The options include **AND** and **OR**. Minimum length: 0 characters Maximum length: 20 characters Enumerated values: • **AND** • **OR**
condition_node	Object	Conditions.
sub_conditions	Array of **RowPermissionDescribe** objects	Row condition list, and the conditions can be nested.

Table 7-8 RowPermissionDescribe

Parameter	Type	Description
logic_operator	String	Relationship between sub-conditions at the current level. The options include **AND** and **OR**. Enumerated values: • **AND** • **OR**
condition_node	**condition_node** object	Conditions.

Parameter	Type	Description
sub_conditions	Array of **RowPermissionDescribe** objects	Row condition list.

Table 7-9 condition_node

Parameter	Type	Description
column_id	String	Field ID.
column_name	String	Field name.
is_measure	Boolean	Whether the field is a metric field.
execute_expression	String	Executable function expression.
relation_operator	String	Condition operator. Note: The content in the following brackets is the explanation of the enumerated values and is not part of the condition operators. The options include **EQUAL-TO** (=), **NOT-EQUAL** (!=), **GREATER-THAN** (>), **GREATER-THAN-OR-EQUAL-TO** (>=), **LESS-THAN** (<), **LESS-THAN-OR-EQUAL-TO** (<=), **""** (without condition operator), **BETWEEN**, **ABSOLUTE** (absolute), **IN**, **NOT-IN**, **START-WITH**, **NOT-START-WITH**, **END-WITH**, **NOT-END-WITH**, **CONTAIN**, **NOT-CONTAIN**, **NULL** (IS NULL), and **NOT-NULL** (IS NOT NULL).
data_type	String	Field type.
value	**value** object	Condition value.

Table 7-10 value

Parameter	Type	Description
values	Array of strings	Value list.
value_type	String	Value type. The options include **ENUM**, **CONDITION**, **TAG_USER_GROUP**, and **TAG_USER**.
value_names	String	List of value names. Mandatory when **value_type** is set to **TAG_USER_GROUP** or **TAG_USER**.

Table 7-11 display_fields

Parameter	Type	Description
user_map	Map<String,String>	User description.
group_map	Map<String,String>	User group description.
field_map	Map<String,String>	Field description.

Example Request

```
GET https://{endpoint}/v1/{project_id}/datasets/{dataset_id}/permissions?
limt=10&offset=0&sort_key=isOpen&sort_dir=desc&permission_type=ROW
```

Example Response

```
{
  "count" : 3,
  "page_data" : [ {
    "id" : "xxxxfdff-xxxb-xxxx-8f52-e15e35axxxxx",
    "name" : "COLUMN",
    "dataset_id" : "xxxxd682-e129-xxxx-ad8e-673f02xxxxxx",
    "is_open" : true,
    "permission_type" : "COLUMN",
    "rule_type" : "MASK",
    "rule_scope" : "ALL",
    "project_id" : "xxxx703a500dxxxxfb3c00f9fxxxxx",
    "workspace_Id" : "xxxx6703a50xxxxx2fb3c00f9fxxxxxx",
    "rule_user" : {
      "users" : [ ],
      "user_groups" : [ ]
    },
    "rule_content" : {
      "last" : 1,
      "first" : 1,
      "mask_type" : "RETAIN_FIRST_N_LAST_M",
      "column_ids" : [ "xxxxe9d6-6xxx-4126-bxxx-36fcxxxxce54.amount" ],
      "special_words" : [ "xxx" ]
    },
    "display_fields" : {
      "user_map" : { },
      "group_map" : { },
      "field_map" : {
        "xxxxe9d6-xxxe-4126-bd51-36fc0ee3ce54.amount" : "amount"
      }
    }
  }, {
    "id" : "xxxxda33-dace-xxxx-8xxx-c6dfaxxxxxf1",
    "name" : "USER_TAG",
    "dataset_id" : "xxxxxd13-cxxx-4xxx-b657-829axxxxx20e",
    "is_open" : true,
    "permission_type" : "ROW",
    "rule_type" : "BY_TAG",
    "rule_scope" : "ALL",
    "rule_user" : {
      "users" : [ ],
      "user_groups" : [ ]
```



```
  },
  "rule_content" : {
    "condition_node" : null,
    "logic_operator" : "AND",
    "sub_conditions" : [ {
      "condition_node" : {
        "value" : {
          "values" : [ "xxxx3355-xxxx-4340-9b5d-0bd856xxxxxx" ],
          "value_type" : "TAG_USER_GROUP",
          "value_names" : [ "userGroup2" ]
        },
        "column_id" : "xxxxx722-b74c-4xxx-9630-43ca65xxxxxx.buyer_name",
        "is_measure" : false,
        "column_name" : "buyer_name",
        "relation_operator" : ""
      },
      "logic_operator" : null
    }, {
      "condition_node" : {
        "value" : {
          "values" : [ "xxxxx355-4ffa-xxxx-9b5d-0bd856xxxxx" ],
          "value_type" : "TAG_USER_GROUP",
          "value_names" : [ "userGroup2" ]
        },
        "column_id" : "xxxxx22-b74c-4752-xxx0-xxxx65d0d12c.buyer_name",
        "is_measure" : false,
        "column_name" : "buyer_name",
        "relation_operator" : ""
      },
      "logic_operator" : null
    } ]
  },
  "display_fields" : null
}, {
  "id" : " xxx14b04-ef3d-4ebc-a9ee-c651f563d67c",
  "name" : "Test",
  "dataset_id" : "xxxx4d13-c2a3-426f-b657-829addaxxxxx",
  "is_open" : true,
  "permission_type" : "ROW",
  "rule_type" : "BY_CONDITION",
  "rule_scope" : "SPECIFIED",
  "rule_user" : {
    "users" : [ "xxxxxxxed1b94534xxxxx3de304axxxx", "xxxx987b3dc4xxxx68b8dff1e7xxxxx",
"774d5axxxxxb4950aabc8651308xxxxx" ],
    "user_groups" : [ ]
  },
  "rule_content" : {
    "condition_node" : null,
    "logic_operator" : "AND",
    "sub_conditions" : [ {
      "condition_node" : {
        "value" : {
          "values" : [ "123" ],
          "value_type" : "CONDITION"
        },
        "column_id" : "xxxx6722-b74c-4752-9xxx-43ca65d0d12xxxxx.buyer_name",
        "is_measure" : false,
        "column_name" : "buyer_name",
        "relation_operator" : "CONTAIN"
      },
      "logic_operator" : null
    }, {
      "condition_node" : null,
      "logic_operator" : "OR",
      "sub_conditions" : [ {
        "condition_node" : {
          "value" : {
            "values" : [ "qwe" ],
            "value_type" : "CONDITION"
```

```
          },
          "column_id" : "xxxx722-b74c-4752-xxxx-43ca65dxxxxx.buyer_name",
          "is_measure" : false,
          "column_name" : "buyer_name",
          "relation_operator" : "CONTAIN"
        },
        "logic_operator" : null
      } ]
    } ]
  },
  "display_fields" : {
    "user_map" : {
      "xxxx987b3dc4xxxxxxb8dff1e73xxxx" : "auth_test",
      "xxxxxac55eeb4950axxxx6513080xxxx" : "chenxiantao",
      "xxxxx57ed1b9453xxxx943de304xxxxx" : "abac_test"
    },
    "group_map" : { },
    "field_map" : null
  }
 } ]
}
```


Status Codes

Table 7-12 Status code

Status Code	Description
200	The dataset permission configuration information is successfully retrieved.

Error Codes

See **Error Codes**.

7.2 Configuring Dataset Permissions

Function

This API is used to configure dataset permissions.

URI

POST /v1/{project_id}/datasets/{dataset_id}/permissions

Table 7-13 URI parameter

Parameter	Mandatory	Type	Description
dataset_id	Yes	String	Dataset ID.
project_id	Yes	String	Project ID.

Request Parameters

Table 7-14 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token.
Content-Type	Yes	String	Request body type and character set, which have fixed values.

Table 7-15 Request body parameters

Parameter	Mandatory	Type	Description
dataset_permissions	Yes	Array of objects	Dataset permissions. For details, see **Table 7-16**.

Table 7-16 dataset_permissions

Parameter	Mandatory	Type	Description
id	Yes	String	Primary key ID.
name	Yes	String	Name.
dataset_id	Yes	String	Dataset ID.
is_open	Yes	Boolean	Whether to enable row and column permissions.
permission_type	Yes	String	Permission type. **ROW** indicates that permissions are restricted by row in a database table, and **COLUMN** indicates that permissions are restricted by column in a database table.
rule_type	Yes	String	Rule type. When **permission_type** is set to **ROW**, the options are **BY_CONDITION** and **BY_TAG**. When **permission_type** is set to **COLUMN**, the options are **FORBID** and **MASK**.

Parameter	Mandatory	Type	Description
rule_scope	Yes	String	Rule range. **ALL** indicates that the rule applies to all users. **ALL_NO** indicates that the rule does not apply to any user. **SPECIFIED** indicates that the rule only applies to specified users. **SPECIFIED_NOT** indicates that the rule does not apply to specified users.
rule_user	Yes	object	Description of the user the rule applies to. For details, see **Table 7-17**.
rule_content	Yes	object	Rule content. For details, see **Table 7-18**.
display_fields	Yes	object	Display fields. For details, see **Table 7-19**.

Table 7-17 rule_user

Parameter	Mandatory	Type	Description
users	No	Array of strings	User.
user_groups	No	Array of strings	User group.

Table 7-18 rule_content

Parameter	Mandatory	Type	Description
last	No	Integer	End position of characters to be processed under custom circumstances.
first	No	Integer	Start position of characters to be processed under custom circumstances.
mask_type	No	String	Data masking type.
column_ids	Yes	Array of strings	List of effective field IDs.
special_words	No	Array of strings	Special character list specified by the data masking type.

Table 7-19 display_fields

Parameter	Mandatory	Type	Description
user_map	No	Map<String,String>	User description.
group_map	No	Map<String,String>	User group description.
field_map	No	Map<String,String>	Field description.

Response Parameters

Table 7-20 Response body parameters

Parameter	Type	Description
message	String	Description.

Example Request

```
POST https://{endpoint}/v1/{project_id}/datasets/{dataset_id}/permissions

{
  " dataset_permissions " : [ {
    "id" : "xxxxfdff-xxxx-43cb-8f52-e15e35axxxxx",
    "name" : "COLUMN",
    "dataset_id" : "1e27d682-xxxx-4589-ad8e-673f02d395f4",
    "is_open" : true,
    "permission_type" : "COLUMN",
    "rule_type" : "MASK",
    "rule_scope" : "ALL",
    "project_id" : "xxxx6703a500xxxx2fb3c00f9fxxxxxx",
    "workspace_Id" : "xxxx6703a50xxxxx2fb3c00f9fxxxxxx",
    "rule_user" : {
      "users" : [ ],
      "user_groups" : [ ]
    },
    "rule_content" : {
      "last" : 1,
      "first" : 1,
      "mask_type" : "RETAIN_FIRST_N_LAST_M",
      "column_ids" : [ "xxxxx9d6-6dae-xxxx-bd51-36fc0exxxxxx.amount" ],
      "special_words" : [ "xxx" ]
    },
    "display_fields" : {
      "user_map" : { },
      "group_map" : { },
      "field_map" : {
        "xxxxx9d6-6dae-4126-xxxx-36fc0exxxxxx.amount" : "amount"
      }
    }
  }, {
    "id" : "xxxxxa33-dace-xxxx-8ddf-c6dfa48xxxxx",
    "name" : "USER_TAG",
    "dataset_id" : "xxxx4d13-c2a3-xxxx-b657-829addaxxxxx",
    "is_open" : true,
    "permission_type" : "ROW",
```



```
    "rule_type" : "BY_TAG",
    "rule_scope" : "ALL",
    "rule_user" : {
      "users" : [ ],
      "user_groups" : [ ]
    },
    "rule_content" : {
      "condition_node" : null,
      "logic_operator" : "AND",
      "sub_conditions" : [ {
        "condition_node" : {
          "value" : {
            "values" : [ "xxxx3355-4ffa-xxxx-9b5d-0bd85668xxxx" ],
            "value_type" : "TAG_USER_GROUP",
            "value_names" : [ "userGroup2" ]
          },
          "column_id" : "xxxx6722-b74c-xxxx-9630-43ca65xxxxxx.buyer_name",
          "is_measure" : false,
          "column_name" : "buyer_name",
          "relation_operator" : ""
        },
        "logic_operator" : null
      }, {
        "condition_node" : {
          "value" : {
            "values" : [ "xxxxx355-4ffa-4340-9b5d-0bd856xxxxx" ],
            "value_type" : "TAG_USER_GROUP",
            "value_names" : [ "userGroup2" ]
          },
          "column_id" : "xxxxx722-b74c-4752-xxxx-43ca65dxxxxx.buyer_name",
          "is_measure" : false,
          "column_name" : "buyer_name",
          "relation_operator" : ""
        },
        "logic_operator" : null
      } ]
    },
    "display_fields" : null
  }, {
    "id" : "xxxxxb04-ef3d-4ebc-a9ee-c651f5xxxxxx",
    "name" : "Test",
    "dataset_id" : "xxxxxd13-c2a3-426f-xxxx-xxxxxdaba20e",
    "is_open" : true,
    "permission_type" : "ROW",
    "rule_type" : "BY_CONDITION",
    "rule_scope" : "SPECIFIED",
    "rule_user" : {
      "users" : [ "xxxxx57ed1b9453xxxxx43de304xxxxx", "xxxxx987b3dcxxxxb68b8dff1e7xxxxx",
"xxxxxac55eeb4950aabc86513080f9ba" ],
      "user_groups" : [ ]
    },
    "rule_content" : {
      "condition_node" : null,
      "logic_operator" : "AND",
      "sub_conditions" : [ {
        "condition_node" : {
          "value" : {
            "values" : [ "123" ],
            "value_type" : "CONDITION"
          },
          "column_id" : "xxxx6722-b74c-xxxx-9630-43ca65dxxxxx.buyer_name",
          "is_measure" : false,
          "column_name" : "buyer_name",
          "relation_operator" : "CONTAIN"
        },
        "logic_operator" : null
      }, {
        "condition_node" : null,
        "logic_operator" : "OR",
```



```
      "sub_conditions" : [ {
        "condition_node" : {
          "value" : {
            "values" : [ "qwe" ],
            "value_type" : "CONDITION"
          },
          "column_id" : "xxxxx722-b74c-xxxx-9630-43ca65dxxxxx.buyer_name",
          "is_measure" : false,
          "column_name" : "buyer_name",
          "relation_operator" : "CONTAIN"
        },
        "logic_operator" : null
      } ]
    } ]
  },
  "display_fields" : {
    "user_map" : {
      "xxxxx987b3dc4xxxx68b8dff1exxxxxx" : "auth_test",
      "xxxxxac55eeb4xxxxabc865130xxxxxx" : "chenxiantao",
      "xxxxx57ed1b94xxxx6c943de30xxxxxx" : "abac_test"
    },
    "group_map" : { },
    "field_map" : null
  }
 } ]
}
```

Example Response

```
{
  "message" : "success"
}
```

Status Codes

Table 7-21 Status code

Status Code	Description
200	The configuration is successful.

Error Codes

See **Error Codes**.

7.3 Enabling or Disabling Dataset Permissions

Function

This API is used to enable or disable dataset permissions.

URI

POST /v1/{project_id}/datasets/{dataset_id}/permissions/config

Table 7-22 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.
dataset_id	Yes	String	Dataset ID.

Request Parameters

Table 7-23 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token.

Table 7-24 Request body parameters

Parameter	Mandatory	Type	Description
col_permission_config	No	object	Column permission configuration. For details, see **Table 7-25**.
row_permission_config	No	object	Row permission configuration. For details, see **Table 7-26**.

Table 7-25 ColPermissionConfig

Parameter	Mandatory	Type	Description
is_open	No	Boolean	Whether to enable column permissions.

Table 7-26 RowPermissionConfig

Parameter	Mandatory	Type	Description
is_open	No	Boolean	Whether to enable row permissions.
is_open_by_condition	No	Boolean	Whether to enable condition combination authorization.

Parameter	Mandatory	Type	Description
is_open_by_tag	No	Boolean	Additional configuration for row and column permissions, indicating whether to enable authorization for combined row conditions.
others_has_permission_by_condition	No	Boolean	Permissions of other users (not specified in the authorization for combined row conditions).

Response Parameters

Table 7-27 Response body parameters

Parameter	Type	Description
data	Boolean	Whether the request is successful.

Example Request

```
POST https://{endpoint}/v1/{project_id}/datasets/{dataset_id}/permissions/config
{
  "row_permission_config" : {
    "is_open" : true
  }
}
```

Example Response

```
{
  "data" : true
}
```

Status Codes

Table 7-28 Status code

Status Code	Description
200	The permissions are successfully updated.

Error Codes

See **Error Codes**.

7.4 Deleting Dataset Permissions

Function

This API is used to delete dataset permissions.

URI

DELETE /v1/{project_id}/datasets/{dataset_id}/permissions/{permission_id}

Table 7-29 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.
dataset_id	Yes	String	Dataset ID.
permission_id	Yes	String	Permission ID.

Request Parameters

Table 7-30 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token.

Response Parameters

Table 7-31 Response body parameters

Parameter	Type	Description
data	Boolean	Whether the request is successful. **true** indicates that the request is successful, and **false** indicates that the request fails.

Example Request

```
DELETE https://{endpoint}/v1/{project_id}/datasets/{dataset_id}/permissions/{permission_id}
```

Example Response

```
{
  "data" : true
}
```

Status Codes

Table 7-32 Status code

Status Code	Description
200	The permissions are successfully deleted.

Error Codes

See **Error Codes**.

7.5 Obtaining Dataset Permission Configuration Information

Function

This API is used to obtain dataset permission configuration information.

URI

GET /v1/{project_id}/datasets/{dataset_id}/permission-config

Table 7-33 URI parameter

Parameter	Mandatory	Type	Description
dataset_id	Yes	String	Dataset ID.
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Request Parameters

Table 7-34 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token.
Content-Type	Yes	String	Request body type and character set, which have fixed values.

Response Parameters

Table 7-35 Response body parameters

Parameter	Type	Description
row_permission_config	object	Row permission configuration. For details, see **Table 7-36**.
col_permission_config	object	Column permission configuration. For details, see **Table 7-37**.

Table 7-36 row_permission_config

Parameter	Type	Description
is_open	Boolean	Whether to enable row permissions.
is_open_by_condition	Boolean	Whether to enable condition combination authorization.
others_has_permission_by_condition	Boolean	Permissions of other users (not specified in the authorization for combined row conditions).
is_open_by_tag	Boolean	Additional configuration for row and column permissions, indicating whether to enable authorization for combined row conditions.

Table 7-37 col_permission_config

Parameter	Type	Description
is_open	Boolean	Whether to enable column permissions.

Example Request

```
GET https://{endpoint}/v1/{project_id}/datasets/{dataset_id}/permission-config
```

Example Response

```
{
  "row_permission_config" : {
    "is_open" : true,
    "is_open_by_condition" : true,
    "others_has_permission_by_condition" : false,
    "is_open_by_tag" : true
  },
  "col_permission_config" : {
    "is_open" : true
  }
}
```

Status Codes

Table 7-38 Status code

Status Code	Description
200	The dataset permission configuration information is successfully retrieved.

Error Codes

See **Error Codes**.

8 Dataset

8.1 Listing Datasets and Directories

Function

This API is used to list datasets.

URI

GET /v1/{project_id}/datasets

Table 8-1 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Table 8-2 Query parameters

Parameter	Mandatory	Type	Description
name	No	String	Dataset name.
sort_dir	No	String	Whether query results are displayed in descending order.
sort_key	No	String	Sorting field.
limit	No	Integer	Number of returned entries. If not transferred, the first 10 records are queried by default. Value range: [0, 1000]. Default value: **10**.

Parameter	Mandatory	Type	Description
offset	No	Integer	Start offset of returned entries. Value range: [0, 1000]. Default value: **0**.

Request Parameters

Table 8-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 8-4 Response body parameters

Parameter	Type	Description
count	Long	Number of data records.
page_data	Array of **Table 8-5** objects	Data.

Table 8-5 DatasetListVO

Parameter	Type	Description
create_date	Long	Creation time.
create_user	String	Creator ID.
create_user_name	String	Creator name.

Parameter	Type	Description
description	String	Description.
id	String	ID.
name	String	Name.
permission_list	Array of strings	Current user's sharing and collaboration permissions. The options are **edit**, **export**, **use**, and **read**.
project_id	String	Project ID.
type	String	Dataset type. The options are: • **table**: dataset consisting of physical tables. • **sql**: dataset that contains custom SQL tables.
update_date	Long	Modification time.
update_user	String	Modifier ID.
update_user_name	String	Modifier name.
workspace_id	String	Workspace ID.
ds_name	String	Name of the bound data source.
ds_type	String	Type of the bound data source. The options are **MYSQL**, **POSTGRESQL**, **DWS**, **OPENGAUSS**, **CLICKHOUSE**, **API**, and **DORIS**.
ds_id	String	ID of the bound data source.
routing_strategy	String	Routing policy for querying data. The options include **Direct** (direct connection) and **Accelerator** (memory engine).
resource_code	String	Dataset resource ID.

Example Request

```
GET https://{endpoint}/v1/{project_id}/datasets?
limit=20&offset=0&sort_dir=desc&name=&create_user_name=&update_user_name=
```

Example Response

Status code: 200

Datasets and directories are successfully listed.

```
{
  "count" : 216,
  "page_data" : [ {
    "id" : "4df79d49-444c-xxxxx-28f9cfeac554",
    "name" : "Save Dataset",
```



```
  "type" : "table",
  "permission_list" : [ "edit", "use" ],
  "workspace_id" : "e77ff0dxxxxxxx1b59fac2cc12faa146f",
  "project_id" : "34ba64e53a02xxxxxxx75669a8cc17bd",
  "create_user" : "558def45cexxxxxxxxa20e463e017",
  "create_user_name" : "xxxxxxxxx",
  "create_date" : 1704533906073,
  "update_user" : "558def4xxxxxxxxx648a20e463e017",
  "update_user_name" : "xxxxxxxx",
  "update_date" : 1704533906073,
  "description" : null,
  "ds_id" : "ff8080818xxxxxf101886aab31a900e1",
  "ds_type" : "DWS",
  "ds_name" : "dws04_public",
  "routing_strategy" : "Direct",
  "resource_code" : "xxx"
 } ]
}
```


Status Codes

Status Code	Description
200	Datasets and directories are successfully listed.

Error Codes

See **Error Codes**.

8.2 Obtaining Dataset Details

Function

This API is used to obtain dataset details.

URI

GET /v1/{project_id}/datasets/{dataset_id}/metadata

Table 8-6 URI parameter

Parameter	Mandatory	Type	Description
dataset_id	Yes	String	Dataset ID.
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Request Parameters

Table 8-7 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 8-8 Response body parameters

Parameter	Type	Description
caption	String	Semantic name of a dataset.
domain_id	String	Tenant ID.
ds_id	String	Data source ID.
ds_type	String	Data source type.
id	String	Dataset ID.
logical_schema	**Table 8-9** object	Logical schema of the dataset.
physical_schema	**Table 8-18** object	Physical schema of the dataset.
project_id	String	Project ID.
version	String	Dataset version. The default value is **2.0**. The options include **1.0** and **2.0**.
workspace_id	String	Workspace ID.
create_date	Long	Creation timestamp.
create_user	String	Creator ID.
create_user_name	String	Creator name.

Parameter	Type	Description
update_date	Long	Update timestamp.
update_user	String	ID of the user who performs the update.
update_user_name	String	Name of the user who performs the update.
resource_code	String	Dataset resource ID.

Table 8-9 LogicalSchema

Parameter	Type	Description
field_schema	**Table 8-10** object	Field schema.
relations	Array of **Table 8-15** objects	Physical table relations.
variables	Array of **Table 8-17** objects	List of SQL variables.

Table 8-10 DatasetFieldSchemaVO

Parameter	Type	Description
dimensions	Array of **Table 8-11** objects	Dimensions.
measures	Array of **Table 8-14** objects	Metrics.

Table 8-11 DimensionVo

Parameter	Type	Description
caption	String	Semantic name of a dimension.
cube_id	String	ID of the physical table a dimension belongs to.
description	String	Dimension description.
id	String	Dimension ID.

Parameter	Type	Description
hierarchies	Array of **Table 8-12** objects	Dimension levels.

Table 8-12 Hierarchy

Parameter	Type	Description
caption	String	Semantic name of a hierarchical structure.
levels	Array of **Table 8-13** objects	Hierarchy levels.

Table 8-13 LevelVO

Parameter	Type	Description
caption	String	Semantic name.
column_formula	String	Level expression.
data_type	String	Field type. The options are **STRING**, **DATE**, and **DATETIME**. Enumerated values: • **STRING|DATE|DATETIME**
expansion_type	Integer	Extension field type. The options are **0** (common extension field) and **1** (aggregation extension field).
id	String	ID.
is_expansion	Integer	Whether a field is an extension field. The options are **0** (no) and **1** (yes). Enumerated values: • **0** • **1**
level_type	String	Level type.
origin_column_name	String	Original field name.
origin_column_type	String	Original field type.

Parameter	Type	Description
origin_data_type	String	Data type before conversion. The options are **STRING**, **DATE**, and **DATETIME**. Enumerated values: • **STRING|DATE|DATETIME**

Table 8-14 MeasureVo

Parameter	Type	Description
caption	String	Semantic name of a metric.
column_formula	String	Field expression of a metric.
cube_id	String	ID of the physical table a field belongs to.
data_type	String	Data type. The options are **STRING**, **NUMBER**, **DATE**, and **DATETIME**. Enumerated values: • **STRING|NUMBER|DATE|DATETIME**
expansion_type	Integer	Extension field type. The options are **0** (common extension field) and **1** (aggregation extension field).
id	String	Metric ID.
is_expansion	Integer	Whether a field is an extension field. The options are **0** (no) and **1** (yes).
origin_column_name	String	Original field name.
origin_column_type	String	Original field type.
origin_data_type	String	Data type before conversion. The options are **STRING**, **NUMBER**, **DATE**, and **DATETIME**. Enumerated values: • **STRING|NUMBER|DATE|DATETIME**

Table 8-15 DatasetRelationVO

Parameter	Type	Description
join_type	String	Join type. The options are **left join**, **right join**, and **inner join**.

Parameter	Type	Description
joins	Array of Table 8-16 objects	Join key.
relation	String	Relation type. The options include **one-to-one**, **one-to-many**, and **many-to-one**.
source	String	Source table ID.
target	String	Target table ID.

Table 8-16 DatasetJoinVO

Parameter	Type	Description
condition	String	Relation. The value is **equal-to**.
source_key	String	Source field ID.
source_type	String	Source field type. The options are **dimension** and **measure**.
target_key	String	Target field ID.
target_type	String	Target field type. The options are **dimension** and **measure**.

Table 8-17 SqlVariable

Parameter	Type	Description
default_values	Array of objects	Default variable value.
format	String	Variable format, which is valid only for variables of the date and time type.
name	String	Variable name.
type	String	Variable type. The options include **STRING**, **NUMBER**, **DATE**, **DATETIME**, and **TIMESTAMP**. Enumerated values: • **STRING|NUMBER|DATE|DATETIME|TIMESTAMP**
use_scope	String	Variable scope. The options are **dataset** and **global**.

Table 8-18 PhysicalSchema

Parameter	Type	Description
tables	Array of **Table 8-19** objects	Physical tables.

Table 8-19 DatasetTableVO

Parameter	Type	Description
id	String	Table ID.
database_name	String	Database.
schema_name	String	Schema name.
table_name	String	Table name.
table_type	String	Dataset type.
sql_text	String	Query SQL.
is_fact_table	Boolean	Whether the table is an endpoint table.

Example Request

```
GET https://{endpoint}/v2/{project_id}/datasets/{dataset_id}/metadata
```

Example Response

Status code: 200

Dataset details are successfully obtained.


```
{
  "create_date" : 1705458279173,
  "create_user" : "cfcxxxx4501811aedcxxxxbbfe8",
  "create_user_name" : "xxxxxxx",
  "update_date" : 1705458279173,
  "update_user" : "cfcc77xxxx364501811aedcxxxxbbfe8",
  "update_user_name" : "xxxxxxx",
  "project_id" : "34ba64e53axxxb68cd7xxxxa8cc17bd",
  "workspace_id" : "d39157xxxx5f43c2bdxxxx74792b2d5e",
  "id" : "b745d57b-xxxx-4f2a-xxxx-3c0ff5fcf92f",
  "caption" : "API Test Dataset 10",
  "version" : "2.0",
  "ds_id" : "2eaa208dxxxxxxx9496914134fae2d4",
  "ds_type" : "DWS",
  "domain_id" : "10ae45e7xxxxxxbe954a211426d003",
  "physical_schema" : {
    "tables" : [ {
      "id" : "a25173cb-xxxx-42ec-xxxxx-67483f6cdaab",
      "database_name" : "modernbi_demo",
      "schema_name" : "autotest",
      "table_name" : "order_info",
      "table_type" : "table",
```



```
    "sql_text" : "",
    "is_fact_table" : false
   }, {
    "id" : "b745d57b-xxxx-4f2a-xxxx-3c0ff5fcf92f",
    "database_name" : "modernbi_demo",
    "schema_name" : "autotest",
    "table_name" : "sql_order_info",
    "table_type" : "sql",
    "sql_text" : "select * from autotest.order_info",
    "is_fact_table" : true
   } ]
  },
  "logical_schema" : {
   "field_schema" : {
    "measures" : [ {
     "id" : "2d83ad17-32bb-xxxx-9c55-xxxxxxx",
     "caption" : "cs_bill_customer_sk",
     "is_expansion" : 0,
     "expansion_type" : 0,
     "column_formula" : "cs_xxxxx_customer_sk",
     "origin_column_name" : "cs_xxxx_customer_sk",
     "origin_column_type" : "int8",
     "data_type" : "NUMBER",
     "origin_data_type" : "NUMBER",
     "cube_id" : "2d83ad17-xxxxxxxxx-9c55-2a47415bdaf6"
    } ],
    "dimensions" : [ {
     "id" : "a25173cb-e229-xxxxxxx-67483f6cdaab.id",
     "caption" : "id1",
     "dimension_type" : "StandardDimension",
     "hierarchies" : [ {
      "caption" : "id1",
      "levels" : [ {
       "id" : "a25173cb-xxxxxxxxc-a4b4-67483f6cdaab.id",
       "caption" : "id1",
       "data_type" : "STRING",
       "origin_data_type" : "STRING",
       "origin_column_name" : "id",
       "origin_column_type" : "varchar(255)",
       "column_formula" : "id",
       "level_type" : null,
       "is_expansion" : 0,
       "expansion_type" : 0
      } ]
     } ],
     "cube_id" : "a25173cb-xxxxxxxc-a4b4-67483f6cdaab"
    } ]
   },
   "variables" : null,
   "relations" : [ {
    "source" : "b745d57b-axxxxx8ee4-3c0ff5fcf92f",
    "target" : "a25173cb-xxxxxx-a4b4-67483f6cdaab",
    "join_type" : "left join",
    "relation" : "many-to-one",
    "joins" : [ {
     "source_key" : "product_id",
     "source_type" : "dimension",
     "target_key" : "product_id",
     "target_type" : "dimension",
     "condition" : "equal-to"
    } ]
   } ]
  }
}
```


Status Codes

Status Code	Description
200	Dataset details are successfully obtained.

Error Codes

See **Error Codes**.

8.3 Deleting a Dataset

Function

This API is used to delete datasets.

URI

DELETE /v1/{project_id}/datasets/{dataset_id}

Table 8-20 URI parameter

Parameter	Mandatory	Type	Description
dataset_id	Yes	String	Dataset ID.
project_id	Yes	String	Project ID. For how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 8-21 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 8-22 Response body parameter

Parameter	Type	Description
data	Boolean	**Explanation** Execution result. **Constraints** N/A **Value range** • **true**: success. • **false**: failure. **Default value** N/A

Example Request

```
DELETE https://{endpoint}/v2/{project_id}/datasets/{dataset_id}
```

Example Response

Status code: 200

The dataset is successfully deleted.

```
{
  "data" : true
}
```

Status Codes

Status Code	Description
200	The dataset is successfully deleted.

Error Codes

See **Error Codes**.

8.4 Saving or Modifying a Dataset

Function

This API is used to save or modify datasets.

URI

POST /v1/{project_id}/datasets/save

Table 8-23 URI parameter

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

Request Parameters

Table 8-24 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 8-25 Request body parameters

Parameter	Mandatory	Type	Description
id	No	String	Dataset ID.
caption	Yes	String	Semantic name of a dataset.
status	No	String	Dataset status. The value can be **0** or **1**.
description	No	String	Dataset description.
ds_id	Yes	String	Data source ID.
ds_name	No	String	Data source name.
ds_type	No	String	Data source type.

Parameter	Mandatory	Type	Description
table_type	Yes	String	Dataset type. The options include: • **table**: dataset that does not contain custom SQL tables. • **sql**: dataset that contains custom SQL tables.
resource_code	No	String	Dataset resource ID.
physical_schema	Yes	**Table 8-26** object	Physical schema of the dataset.
logical_schema	Yes	**Table 8-28** object	Logical schema of the dataset.

Table 8-26 PhysicalOpenApiSchema

Parameter	Mandatory	Type	Description
tables	Yes	Array of **Table 8-27** objects	Physical tables.

Table 8-27 DatasetTableOpenApiVO

Parameter	Mandatory	Type	Description
database_name	Yes	String	Database.
is_fact_table	Yes	Boolean	Whether the table is an endpoint table.
schema_name	No	String	Schema.
sql_text	No	String	Query SQL.
table_name	Yes	String	Table name.
table_type	Yes	String	Table type. The options include: • **table**: physical table. • **sql**: custom SQL table.

Table 8-28 LogicalSchemaOpenApiVO

Parameter	Mandatory	Type	Description
filed_schema	No	**Table 8-29** object	Field selection information.
variables	No	Array of **Table 8-31** objects	List of SQL variables.
relations	No	Array of **Table 8-32** objects	Physical table relations.

Table 8-29 DatasetFieldSchemaOpenApiVO

Parameter	Mandatory	Type	Description
columns	No	Array of **Table 8-30** objects	Field selection list and field name assignment.

Table 8-30 ColumnOpenApiVO

Parameter	Mandatory	Type	Description
caption	Yes	String	Field name.
origin_column_name	Yes	String	Physical name of a field, meaning the name in the table.
schema_name	No	String	Field schema.
table_name	Yes	String	Name of the table where the field is.

Table 8-31 SqlVariable

Parameter	Mandatory	Type	Description
default_values	No	Array of objects	Default variable value.
format	No	String	Variable format, which is valid only for variables of the date and time type.

Parameter	Mandatory	Type	Description
name	Yes	String	Variable name.
type	Yes	String	Variable type. The options include **STRING**, **NUMBER**, **DATE**, **DATETIME**, and **TIMESTAMP**. Enumerated values: • **STRING|NUMBER|DATE|DATETIME|TIMESTAMP**
use_scope	Yes	String	Variable scope. The options are **dataset** and **global**.

Table 8-32 DatasetRelationOpenApiVO

Parameter	Mandatory	Type	Description
source_database_name	Yes	String	Name of the database the source table belongs to.
source_schema	No	String	Name of the schema the source table belongs to.
source_table_name	Yes	String	Source table name.
target_database_name	Yes	String	Name of the database the target table belongs to.
target_schema	No	String	Name of the schema the target table belongs to.
target_table_name	Yes	String	Target table name.
join_type	Yes	String	Join type.
relation	Yes	String	Relation type.
joins	Yes	Array of **Table 8-33** objects	Join key.

Table 8-33 DatasetJoinOpenApiVO

Parameter	Mandatory	Type	Description
condition	Yes	String	Join relationship. The default value is **equal-to**.

Parameter	Mandatory	Type	Description
source_key	Yes	String	Source table field.
target_key	Yes	String	Target table field.

Response Parameters

Status code: 200

Table 8-34 Response body parameters

Parameter	Type	Description
caption	String	Semantic name of a dataset.
domain_id	String	Tenant ID.
ds_id	String	Data source ID.
ds_type	String	Data source type.
id	String	Dataset ID.
logical_schema	Table 8-35 object	Logical schema of the dataset.
physical_schema	Table 8-44 object	Physical schema of the dataset.
project_id	String	Project ID.
version	String	Dataset version. The default value is **2.0**. The options include **1.0** and **2.0**.
workspace_id	String	Workspace ID.
create_date	Long	Creation timestamp.
create_user	String	Creator ID.
create_user_name	String	Creator name.
update_date	Long	Update timestamp.
update_user	String	ID of the user who performs the update.
update_user_name	String	Name of the user who performs the update.
resource_code	String	Dataset resource ID.

Table 8-35 LogicalSchema

Parameter	Type	Description
field_schema	**Table 8-36** object	Field schema.
relations	Array of **Table 8-41** objects	Physical table relations.
variables	Array of **Table 8-43** objects	List of SQL variables.

Table 8-36 DatasetFieldSchemaVO

Parameter	Type	Description
dimensions	Array of **Table 8-37** objects	Dimensions.
measures	Array of **Table 8-40** objects	Metrics.

Table 8-37 DimensionVo

Parameter	Type	Description
caption	String	Semantic name of a dimension.
cube_id	String	ID of the physical table a dimension belongs to.
description	String	Dimension description.
id	String	Dimension ID.
hierarchies	Array of **Table 8-38** objects	Dimension levels.

Table 8-38 Hierarchy

Parameter	Type	Description
caption	String	Semantic name of a hierarchical structure.

Parameter	Type	Description
levels	Array of Table 8-39 objects	Hierarchy levels.

Table 8-39 LevelVO

Parameter	Type	Description
caption	String	Semantic name.
column_formula	String	Level expression.
data_type	String	Field type. The options are **STRING**, **DATE**, and **DATETIME**. Enumerated values: • **STRING|DATE|DATETIME**
expansion_type	Integer	Extension field type. The options are **0** (common extension field) and **1** (aggregation extension field).
id	String	ID.
is_expansion	Integer	Whether a field is an extension field. The options are **0** (no) and **1** (yes). Enumerated values: • **0** • **1**
level_type	String	Level type.
origin_column_name	String	Original field name.
origin_column_type	String	Original field type.
origin_data_type	String	Data type before conversion. The options are **STRING**, **DATE**, and **DATETIME**. Enumerated values: • **STRING|DATE|DATETIME**

Table 8-40 MeasureVo

Parameter	Type	Description
caption	String	Semantic name of a metric.

Parameter	Type	Description
column_formula	String	Field expression of a metric.
cube_id	String	ID of the physical table a field belongs to.
data_type	String	Data type. The options are **STRING**, **NUMBER**, **DATE**, and **DATETIME**. Enumerated values: • **STRING|NUMBER|DATE|DATETIME**
expansion_type	Integer	Extension field type. The options are **0** (common extension field) and **1** (aggregation extension field).
id	String	Metric ID.
is_expansion	Integer	Whether a field is an extension field. The options are **0** (no) and **1** (yes).
origin_column_name	String	Original field name.
origin_column_type	String	Original field type.
origin_data_type	String	Data type before conversion. The options are **STRING**, **NUMBER**, **DATE**, and **DATETIME**. Enumerated values: • **STRING|NUMBER|DATE|DATETIME**

Table 8-41 DatasetRelationVO

Parameter	Type	Description
join_type	String	Join type. The options are **left join**, **right join**, and **inner join**.
joins	Array of **Table 8-42** objects	Join key.
relation	String	Relation type. The options include **one-to-one**, **one-to-many**, and **many-to-one**.
source	String	Source table ID.
target	String	Target table ID.

Table 8-42 DatasetJoinVO

Parameter	Type	Description
condition	String	Relation. The value is **equal-to**.
source_key	String	Source field ID.
source_type	String	Source field type. The options are **dimension** and **measure**.
target_key	String	Target field ID.
target_type	String	Target field type. The options are **dimension** and **measure**.

Table 8-43 SqlVariable

Parameter	Type	Description
default_values	Array of objects	Default variable value.
format	String	Variable format, which is valid only for variables of the date and time type.
name	String	Variable name.
type	String	Variable type. The options include **STRING**, **NUMBER**, **DATE**, **DATETIME**, and **TIMESTAMP**. Enumerated values: • **STRING|NUMBER|DATE|DATETIME|TIMESTAMP**
use_scope	String	Variable scope. The options are **dataset** and **global**.

Table 8-44 PhysicalSchema

Parameter	Type	Description
tables	Array of **Table 8-45** objects	Physical tables. Array length: 0 to 2,147,483,647 elements

Table 8-45 DatasetTableVO

Parameter	Type	Description
id	String	Table ID.

Parameter	Type	Description
database_name	String	Database.
schema_name	String	Schema name.
table_name	String	Table name.
table_type	String	Dataset type.
sql_text	String	Query SQL.
is_fact_table	Boolean	Whether the table is an endpoint table.

Example Request

```
POST https://{endpoint}/v1/{project_id}/datasets/save
{
  "id" : "string",
  "caption" : "id",
  "status" : "string",
  "description" : "string",
  "ds_id" : "string",
  "ds_name" : "string",
  "ds_type" : "string",
  "table_type" : "table|sql",
  "resource_code" : "string",
  "physical_schema" : {
    "tables" : [ {
      "database_name" : "db_test",
      "is_fact_table" : false,
      "schema_name" : "public",
      "sql_text" : "SELECT * FROM FACT_GDP",
      "table_name" : "t_test",
      "table_type" : "table"
    } ]
  },
  "logical_schema" : {
    "field_schema" : {
      "columns" : [ {
        "caption" : "age",
        "origin_column_name" : "age",
        "schema_name" : "public",
        "table_name" : "modern_bi_0xxxx1c590b14d9xxxx35227166f7f02"
      } ]
    },
    "variables" : [ {
      "default_values" : [ "aaa", "bbb" ],
      "format" : "yyyy-MM-dd",
      "name" : "limit",
      "type" : "STRING",
      "use_scope" : "dataset"
    } ],
    "relations" : [ {
      "source_database_name" : "string",
      "source_schema" : "string",
      "source_table_name" : "string",
      "target_database_name" : "string",
      "target_schema" : "string",
      "target_table_name" : "string",
      "join_type" : "string",
      "relation" : "string",
      "joins" : [ {
        "condition" : "equal-to",
```



```
      "source_key" : "city",
      "target_key" : "city"
    } ]
  } ]
 }
}
```

Example Response

None

Status Codes

Status Code	Description
200	The dataset is successfully saved or modified.

Error Codes

See **Error Codes**.

9 Embedded Analytics

9.1 Obtaining Shared Embedded Analytics Information

Function

This API is used to obtain shared embedded analytics information.

URI

GET /v1/{project_id}/embeddings/shared/{embed_id}/attributes

Table 9-1 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.
embed_id	Yes	String	Embedded analytics ID.

Request Parameters

Table 9-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 9-3 Response body parameters

Parameter	Type	Description
resource_id	String	Resource ID.
root_resource_id	String	Root ID.
resource_type	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
workspace_id	String	Workspace ID.

Example Request

```
GET https://{host}/v1/{project_id}/embeddings/shared/{embed_id}/attributes
```

Example Response

Status code: 200

```
{
  "root_resource_id" : "0cc302c0-xxxxxxx-8bdf-0f726fb36eb4",
  "resource_type" : "screen",
  "resource_id" : "2befed28-xxxxx-83df-dbc88e47ee12",
  "workspace_id" : "b8cbb94cce8xxxxxxx7d8e5498af5dd"
}
```


Status Codes

Status Code	Description
200	The request parameters are successfully uploaded.

Error Codes

See **Error Codes**.

9.2 Creating Embedded Analytics

Function

This API is used to create embedded analytics.

URI

POST /v1/{project_id}/embeddings

Table 9-4 URI parameter

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

Request Parameters

Table 9-5 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 9-6 Request body parameters

Parameter	Mandatory	Type	Description
id	No	String	Embedded analytics ID. If specified, the ID that is passed in is used. Or, a new ID will be generated.
name	Yes	String	Embedded analytics name.
resource_type	Yes	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
resource_id	Yes	String	Resource ID corresponding to the embedded analytics, for example, the ID of the dashboard or large screen.
root_resource_id	Yes	String	Root resource ID corresponding to the embedded analytics, which is the root resource ID of the dashboard or large screen.

Parameter	Mandatory	Type	Description
auth_type	No	String	Access type of the embedded analytics. **0** indicates login access, and **1** indicates ticket authentication. The default value is **0**.
config	No	String	Configuration parameter of the embedded analytics. For example, **ticket=xxxx** indicates the ticket value in the embedded analytics link.

Response Parameters

Status code: 200

Table 9-7 Response body parameter

Parameter	Type	Description
id	String	ID of the created embedded analytics.

Example Request

Create embedded analytics.

```
POST https://{endpoint}/v1/{project_id}/embeddings

{
  "id" : "6cb08a50xxxxxxxxxxxxxxxxxxxxxxx",
  "name": "New Embedded Analytics",
  "resource_type" : "dashboard",
  "resource_id" : "866a519fxxxxxxxxxxxxxxxx",
  "root_resource_id" : "866a519fxxxxxxxxxxxxxxxx",
  "config" : "ticket=ec85f874axxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "auth_type" : 1
}
```

Example Response

Status code: 200

The embedded analytics is created.

```
{
"id": "6cb08a50xxxxxxxxxxxxxxxxxxxxxxx",
}
```


Status Codes

Status Code	Description
200	The embedded analytics is created.

Error Codes

See **Error Codes**.

10 Ticket Management

10.1 Creating a Ticket

Function

This API is used to create a ticket.

URI

POST /v1/{project_id}/tickets

Table 10-1 URI parameter

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

Request Parameters

Table 10-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 10-3 Request body parameters

Parameter	Mandatory	Type	Description
embed_id	Yes	String	Embedded analytics ID.
user_id	Yes	String	User ID.
group_ids	No	Array of strings	List of user group IDs.
ticket_num	No	Integer	Number of times the ticket has been used.
session_auto_renew	No	Boolean	Whether to automatically renew the session. Default value: **false**

Response Parameters

Status code: 200

Table 10-4 Response body parameter

Parameter	Type	Description
ticket	String	Generated ticket.

Example Request

Create a ticket.

```
POST https://{endpoint}/v1/{project_id}/tickets

{
  "user_id" : "aaa18eb155xxxxxxxxxxxxxxxxxxx",
  "ticket_num" : 1000,
  "embed_id" : "8e59605cxxxxxxxxxxxx"
}
```

Example Response

Status code: 200

The ticket is created.

```
{
  "ticket" : "dea43d00388xxxxxxxxxxxxxx"
}
```


Status Codes

Status Code	Description
200	The ticket is created.

Error Codes

See **Error Codes**.

10.2 Listing Tickets

Function

This API is used to list valid tickets.

URI

GET /v1/{project_id}/tickets

Table 10-5 URI parameter

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

Table 10-6 Query parameters

Parameter	Mandatory	Type	Description
user_id	No	String	User ID.
user_name	No	String	Username.
user_type	No	Integer	User type.
embed_id	No	String	Embedded analytics ID.
limit	No	Integer	Number of returned entries. If not transferred, the first 10 records are queried by default. Value range: [0, 1000]. Default value: **10**.
offset	No	Integer	Start offset of returned entries. Value range: [0, 1000]. Default value: **0**.

Parameter	Mandatory	Type	Description
sort_dir	No	String	Whether query results are displayed in descending order.
sort_key	No	String	Sorting field.

Request Parameters

Table 10-7 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 10-8 Response body parameters

Parameter	Type	Description
count	Integer	Total number of data records.
page_data	Array of Table 7-9 objects	Data on the current page.

Table 10-9 TicketVo

Parameter	Type	Description
ticket	String	Ticket.
embed_id	String	ID of the embedded analytics to which the ticket is bound.
expire_time	String	Ticket expiration time.
left_num	String	Number of remaining times that the ticket can be used.
user_id	String	ID of the user to whom the ticket is bound.

Parameter	Type	Description
user_name	String	Name of the user to whom the ticket is bound.
user_type	String	Type of the user to which the ticket is bound.
group_ids	Array of strings	ID of the user group to which the ticket is bound.

Example Request

Request ticket list.

```
GET https://{endpoint}/v1/{project_id}/tickets
```

Example Response

Status code: 200

The ticket list is obtained.

```
{
  "count": 2,
  "page_data": [
    {
      "ticket": "848803f514dxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "embed_id": "3577e712-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "expire_time": "2024-08-26 20:48:52 (GMT+08:00)",
      "left_num": 1000,
      "user_id": "aaa18eb1xxxxxxx",
      "user_name": "user_a",
      "user_type": "IAM",
      "group_ids": []
    },
    {
      "ticket": "22a01f2f8cxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "embed_id": "3577e712xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "expire_time": "2024-08-26 20:48:45 (GMT+08:00)",
      "left_num": 1000,
      "user_id": "16246eaxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "user_name": "user_b",
      "user_type": "IAM",
      "group_ids": []
    }
}
```

Status Codes

Status Code	Description
200	The ticket list is obtained.

Error Codes

See **Error Codes**.

11 Import and Export Parameters

11.1 Setting Import and Export Parameters

Function

This API is used to set import and export parameters.

URI

POST /v1/{project_id}/exports/config

Table 11-1 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Request Parameters

Table 11-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token.
Content-Type	Yes	String	Request body type and character set, which have fixed values.

Table 11-3 Request body parameters

Parameter	Mandatory	Type	Description
service_type	Yes	String	Service type.
obs_endpoint	Yes	String	OBS endpoint.
obs_bucket_name	Yes	String	OBS bucket name.
max_storage_days	Yes	Integer	Maximum storage duration of exported files in OBS.
obs_path_prefix	Yes	String	OBS path prefix.

Response Parameters

Status code: 200

Table 11-4 Response body parameters

Parameter	Type	Description
message	String	Description.

Example Request

```
POST https://{endpoint}/v1/{project_id}/exports/config

{
  "service_type" : "obs",
  "obs_endpoint" : "obs.xxx.com",
  "obs_bucket_name" : "xxxxx",
  "max_storage_days" : 1,
  "obs_path_prefix" : "/dataArts-insight"
}
```

Example Response

```
{
  "message" : "success"
}
```


Status Codes

Table 11-5 Status code

Status Code	Description
200	The configuration is successful.

Error Codes

See **Error Codes**.

12 Product Instances

12.1 Listing a User's Enabled Product Instances

Function

This API is used to list a user's enabled product instances.

URI

GET /v1/{project_id}/instances

Table 12-1 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Table 12-2 Query parameters

Parameter	Mandatory	Type	Description
offset	Yes	Integer	Offset of pagination. The value **0** indicates no offset, that is, the first page. The value **1** indicates that the offset is 1, that is, the second page.
limit	Yes	Integer	Maximum number of returned items allowed.

Parameter	Mandatory	Type	Description
sort_key	No	String	Sorting field. The value is **isOpen**, indicating that whether to enable row and column permissions.
sort_dir	No	String	Sorting type. The options are **desc** and **asc**. **desc** indicates the descending order, and **asc** indicates the ascending order.

Request Parameters

Table 12-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token.
Content-Type	Yes	String	Request body type and character set, which have fixed values.

Response Parameters

Table 12-4 Response body parameters

Parameter	Type	Description
instances	Array of objects	Instance information. For details, see **Table 12-5**.
count	Integer	Number of data records.
page_data	Array of objects	Data on the current page.

Table 12-5 instances

Parameter	Type	Description
project_id	String	Project ID.

Parameter	Type	Description
region_id	String	Region ID.
order_id	String	Order ID.
instance_id	String	Product instance ID.
instance_name	String	Product instance name.
resource_spec_code	String	Product specifications
order_type	String	Order type. The options include: • **ON_DEMAND**: Pay-per-use. • **PERIOD**: Yearly/Monthly.
status	Integer	Instance status. The options are: • **1**: Not effective. • **2**: In effect. • **3**: Deleted (unsubscribed). • **4**: In pretension period (frozen). • **5**: In grace period.
domain_id	String	Account ID.
eps_id	String	Enterprise project ID.

Example Request

```
GET https://{endpoint}/v1/{project_id}/instances
```

Example Response

Enabled product instances are successfully listed.

```
{
  "count": 1,
  "page_data": [
    {
      "project_id": "9c3043ab4xxxx055888643b331a0xxxx",
      "region_id": "xxx",
      "order_id": "4xxxxd9517314ebbbe17258a9640xxxx",
      "instance_id": "4e7xxxx517314exxxx17258a9640c05f",
      "instance_name": "xxxx-4e7a0d9517314exxxx17258a9640c05f",
      "instance_spec_code": "DataArtsInsight.unit.enterprise.user",
      "order_type": "ON_DEMAND",
      "status": 2,
      "create_user": "***",
      "create_date": 169xxxx2604761,
      "domain_id": "d22f8acfxxxx428fb22606b64dc7xxxx",
      "eps_id": "0"
    }
  ]
}
```


Status Codes

Status Code	Description
200	Enabled product instances are successfully listed.

Error Codes

See **Error Codes**.

13 Resource Migration

13.1 Exporting a Resource Package Using an API

Function

This API is used to export resource packages.

URI

POST /v1/{project_id}/resource-package/export

Table 13-1 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Request Parameters

Table 13-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token.

Table 13-3 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the exported package.
export_param	No	**ResourcePackageExportParam** object	Exported advanced parameters.
resources	Yes	Array of objects	Details about exported resources.

Table 13-4 ResourcePackageExportParam

Parameter	Mandatory	Type	Description
export_dataset_tag_permission	No	Boolean	Whether the resource package includes dataset row permissions and tag-based authorization.

Table 13-5 resources

Parameter	Mandatory	Type	Description
resource_type	Yes	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
resource_id	Yes	String	Resource ID.

Response Parameters

Table 13-6 Response body parameter

Parameter	Type	Description
-	File	File stream.

Example Request

```
POST https://{endpoint}/v1/{project_id}/resource-package/export
{
  "name": "Unnamed Resource Package",
  "resources" : [ {
    "resource_type" : "datasource",
```



```
    "resource_id" : "12awedqweqweqwe"
  } ]
}
```

Example Response

None

Status Codes

Table 13-7 Status code

Status Code	Description
200	File stream.

Error Codes

See **Error Codes**.

13.2 Obtaining Import Task Details

Function

This API is used to obtain details about import tasks.

URI

GET /v1/{project_id}/resource-package/import-tasks/{task_id}

Table 13-8 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.
task_id	Yes	String	ID of an import task.

Request Parameters

Table 13-9 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Table 13-10 Response body parameters

Parameter	Type	Description
id	String	Task ID.
name	String	Import name.
domain_id	String	Domain ID.
project_id	String	Project ID.
workspace_id	String	Workspace ID.
import_param	String	List of import parameters.
obs_bucket	String	OBS bucket name.
obs_object_key	String	OBS object path.
import_detail	String	Import details.

Parameter	Type	Description
status	Integer	Result status. The options include **-1** (not activated), **0** (to be processed), **1** (successful), **2** (failed), **3** (partially successful), and **4** (being processed). Enumerated values: • **-1** • **0** • **1** • **2** • **3** • **4**
error_msg	String	Task failure cause.
import_type	String	Import type. The options include **import** (resource package import) and **crossSpaceImport** (cross-workspace one-click import).
create_user	String	Creator ID.
create_user_name	String	Creator name.
create_date	Long	Creation timestamp.
update_date	Long	Update timestamp.

Example Request

```
GET https://{endpoint}/v1/{project_id}/resource-package/import-tasks/{task_id}
```

Example Response

Import task details are successfully obtained.

```
{
  "id" : "2406a337-xxxx-a5b2-b01a61d2043b",
  "name" : "string",
  "domain_id" : "ad9e2f08-xxxx-87eb-1f9619b6c313",
  "project_id" : "string",
  "workspace_id" : "12913c92-xxxx-b8fd-a72af3945b61",
  "import_param" : "",
  "obs_bucket" : "test",
  "obs_object_key" : "test.zip",
  "import_detail": "Import successful. [Export resource list] Data source: 1; [Import parameter list]: Target workspace: sample workspace; Replace: false; Carry permission information: false",
  "status" : 1,
  "error_msg" : "string",
  "import_type" : "import|crossSpaceImport",
  "create_user" : "string",
  "create_user_name" : "string",
  "create_date" : 9223372036854776000,
```



```
  "update_date" : 9223372036854776000
}
```

Status Codes

Table 13-11 Status code

Status Code	Description
200	Import task details are successfully obtained.

Error Codes

See **Error Codes**.

13.3 Importing Resource Packages

Function

This API is used to import resource packages.

URI

POST /v1/{project_id}/resource-package/api-import

Table 13-12 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Request Parameters

Table 13-13 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token.

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Request body type and character set, which have fixed values.

Table 13-14 FormData parameters

Parameter	Mandatory	Type	Description
file	Yes	File	Resource package file.
import_strategy	Yes	String	Import strategy. The options are **replace** and **addNew**.
replace_rule	No	String	Replacement rule. The default value is **BY_ID**. The options are **BY_ID** and **BY_NAME**. Enumerated values: • **BY_ID** • **BY_NAME**

Response Parameters

Table 13-15 Response body parameter

Parameter	Type	Description
task_id	String	Task ID.

Example Request

Import a resource package.

```
POST https://{endpoint}/v1/{project_id}/resource-package/api-import
{
    "file": File stream,
    "import_strategy": "replace",
    "replace_rule": "BY_ID"
}
```

Example Response

The resource package is successfully imported.

```
{
  "task_id" : "b8913c92-xxxx-4721-b8fd-a72af3945b61"
}
```


Status Codes

Table 13-16

Status Code	Description
200	The resource package is successfully imported.

Error Codes

See **Error Codes**.

14 Common Resource Query

14.1 Listing Dashboards or Large Screens

Function

This API is used to list dashboards or large screens.

URI

GET /v1/{project_id}/resources/{resource_type}

Table 14-1 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
resource_type	Yes	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A

Table 14-2 Query parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset.
limit	No	Integer	Maximum number of returned items allowed.
sort_key	No	String	Sorting field.
sort_dir	No	String	Sorting type. The options include: • **desc**: descending order. • **asc**: ascending order.

Request Parameters

Table 14-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 14-4 Response body parameters

Parameter	Type	Description
count	Integer	Total number.
page_data	Array of **Table 14-5** objects	Pagination data.

Table 14-5 ResourceVO

Parameter	Type	Description
id	String	ID.
name	String	Resource name.
create_user_name	String	Creator.
create_date	Integer	Creation date.
update_user_name	String	User who performs the update.
update_date	Integer	Update date.
status	Integer	Publish status.

Example Request

List large screens.

```
GET https://{endpoint}/v1/0e****127d****6b7****95750da/resources/screen
```

Example Response

Status code: 200

Resources are successfully listed.

```
{
  "count" : 1,
  "page_data" : [ {
    "id" : "e2xxxx66f-81bxxxxxx9c38-80xxx0fxx9c84",
    "name": "Unnamed Large Screen 1",
    "create_user_name" : "dat***ig***",
    "create_date" : 1707215155901,
    "update_user_name" : "dat***ig***",
    "update_date" : 1707215155901,
    "status" : 0
  } ]
}
```

Status Codes

Status Code	Description
200	Resources are successfully listed.

Error Codes

See **Error Codes**.

14.2 Querying Resource Lineage

Function

This API is used to query resource lineage.

URI

POST /v1/{project_id}/resources/relationships

Table 14-6 URI parameter

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

Request Parameters

Table 14-7 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 14-8 Request body parameters

Parameter	Mandatory	Type	Description
resources	Yes	Array of **resources** objects	List of resources whose lineage is to be queried.

Table 14-9 resources

Parameter	Mandatory	Type	Description
resource_type	Yes	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
resource_id	Yes	String	Resource ID.

Response Parameters

Status code: 200

Table 14-10 Response body parameters

Parameter	Type	Description
resources	Array of **ResourceRelationDetailVO** objects	Resource lineage list.
workspace_id	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.

Table 14-11 ResourceRelationDetailVO

Parameter	Type	Description
resource_id	String	Resource ID.
resource_name	String	Resource name.
resource_type	String	**Explanation** Enumerated value of the resource type, which is used to specify the type of the authorized object. For example, **datasource** indicates that the type of the authorized object is a data source. **Constraints** N/A **Value range** The enumerated values are: • **datasource**: data source. • **dataset**: dataset. • **metric**: indicator. • **dashboard**: dashboard. • **screen**: large screen. • **embedAnalysis**: embedded analytics. • **portal**: data portal. • **subject**: intelligent analysis assistant. **Default value** N/A
relations	Array of **ResourceRelationDetailVO** objects	Dependent resource list.

Example Request

Query the lineage of resources.

```
POST https://{endpoint}/v1/0e****127d****6b7****95750da/resources/relationships
{
  "resources" : [ {
    "resource_type" : "portal",
    "resource_id" : "96334c9****4510a0035dff29a7582b"
  } ]
}
```

Example Response

Status code: 200

The lineage of resources is successfully queried.


```
{
  "resources" : [ {
    "resource_id" : "96334c9923****0a0035dff29a7582b",
    "resource_name": "Test Data Portal",
    "resource_type" : "portal",
    "relations" : [ {
      "resource_id" : "2545c456-****-a6d4-dd8419cac54a",
      "resource_name": "Large Screen",
      "resource_type" : "screen",
      "relations" : [ ]
    }, {
      "resource_id" : "12e25f5f-4*****-8468-bdf69c25438f",
      "resource_name": "Dashboard",
      "resource_type" : "dashboard",
      "relations" : [ {
        "resource_id" : "980bc5e2-******4-a168-4d344065bcb3",
      "resource_name": "Dataset",
        "resource_type" : "dataset",
        "relations" : [ {
          "resource_id" : "c4d84a8******ba190c5cabf75dfe",
          "resource_name" : "datasource",
          "resource_type" : "datasource",
          "relations" : [ ]
        } ]
      } ]
    } ]
  } ],
  "workspace_id" : "e10aa3****e713d779c65e74"
}
```


Status Codes

Status Code	Description
200	The lineage of resources is successfully queried.

Error Codes

See **Error Codes**.

15 Data Source

15.1 Updating a Data Source

Function

This API is used to update data sources.

URI

PUT /v1/{project_id}/connections/{connection_id}

Table 15-1 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see **Obtaining a Project ID**.
connection_id	Yes	String	Data source ID.

Request Parameters

Table 15-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 15-3 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Data source name.
description	No	String	Data source description.
type	Yes	String	**Explanation** Data source type. For example, **MySQL** indicates that the type of the connected data source is MySQL. The MySQL driver package is used for query. **Constraints** N/A **Value range** The enumerated values are: • **MySQL**. • **DWS**: data source type corresponding to GaussDB(DWS). • **PostgreSQL**. • **OpenGauss**: data source type corresponding to GaussDB. • **Doris**. • **HIVE**. • **ClickHouse**. • **API**: data source type corresponding to API data. **Default value** N/A

Parameter	Mandatory	Type	Description
source	Yes	String	Data source. Enumerated values: • **rds**: RDS data source. • **dws**: GaussDB(DWS) data source. • **mrs**: MRS data source. • **gaussdb**: GaussDB data source. • **public**: public network data source. • **restapi**: API data source. • **endpoint-service**: VPCEP data source.
server_list	No	String	List of servers. Mandatory if the data source is in cluster mode (containing multiple instances), for example, **DWS** or **ClickHouse**.
database_name	Yes	String	Database name.
host	No	String	Domain name. Mandatory for non-cluster data sources, for example, **mysql**.
port	No	Integer	Port.
user_name	No	String	Username.
password	No	String	Password.
cluster	No	**ClusterInfoDTO** object	Mandatory when any cloud instance is connected.
region_id	No	String	Current region ID.
config	Yes	**DcConfigDTO** object	Configuration parameter.

Table 15-4 ClusterInfoDTO

Parameter	Mandatory	Type	Description
id	No	String	ID of the associated cloud instance.

Table 15-5 DcConfigDTO

Parameter	Mandatory	Type	Description
ssl	No	Boolean	SSL parameter configuration.
region_id	No	String	ID of the region where DLI is. Currently, cross-region deployment is not supported. The value must be the ID of the current region.
catalog_name	No	String	Data directory name. The default value is **dli**. The value of DLI catalogs is **dli**.
catalog_type	No	String	Data directory type. The default value is **dil**. The options include: • **dli**: DLI catalogs. • **lakeformation**: LakeFormation data directories.
ds_project_id	No	String	ID of the project the DLI queue belongs to.
queue_name	No	String	SQL queue.
auth_type	No	String	Authentication mode. The value is **AKSK**. Enumerated values: • **AKSK**
ds_ak	No	String	User AK. Mandatory when the authentication mode is **AKSK**.
ds_sk	No	String	User SK. Mandatory when the authentication mode is **AKSK**.
time_out	No	Integer	Request timeout interval when an API is used as the data source.
protocol	No	String	Protocol. The value is **REST**.
token_key	No	String	Token key.
token_value	No	String	Token value.
request_url	No	String	Request URL when an API is used as the data source.

Parameter	Mandatory	Type	Description
request_method	No	String	Request method. The value can be **GET** or **POST**. Enumerated values: • **GET** • **POST**
request_params	No	Array of **RequestParamDto** objects	List of request parameters.
resp_param	No	Array of **ResponseParamDto** objects	Response parameters.

Table 15-6 RequestParamDto

Parameter	Mandatory	Type	Description
location	Yes	String	Location.
value	Yes	Array of strings	Parameter value.
tag_id	No	String	Tag ID. Mandatory when **param_source** is set to **user**.
param_name	Yes	String	Parameter name.
alias_name	No	String	Parameter alias.
field_type	Yes	String	Parameter type. The options include **String**, **Long**, **Decimal**, **List<String>**, **List<Long>**, **List<Decimal>**, **DateTime**, **Bool**, **PageIndex**, and **PageNum**.
param_source	Yes	String	Parameter source. The options include **custom**, **user**, and **userGroup**.
type	Yes	Boolean	Whether to encrypt data.

Table 15-7 ResponseParamDto

Parameter	Mandatory	Type	Description
alias_name	Yes	String	Parameter alias.
field_type	Yes	String	Parameter type. The options include **String**, **Long**, **Decimal**, **DateTime**, and **Bool**.
param_name	Yes	String	Parameter name.

Response Parameters

Status code: 200

Table 15-8 Response body parameter

Parameter	Type	Description
message	String	Description.

Example Request

Update data sources.

```
PUT /v1/{project_id}/connections/{connection_id}
Change the data source to a public network connection.
{
  "name": "dws02",
  "server_list": "xx.xx.x.xx:xxxx",
  "database_name": "gaussdb",
  "user_name": "xxxxx",
  "password": "xxxxxxx",
  "config": {
    "ssl": true
  },
  "source": "public",
  "type": "DWS"
}
Change the data source to GaussDB(DWS).
{
    "name": "dws04_instance",
    "description": "Added for Test",
    "source": "dws",
    "type": "DWS",
    "database_name": "xxxxx",
    "user_name": "xxxx",
    "password": "xxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
Change the data source to RDS.
{
```



```
    "name": "mysql_instance",
    "description": "Added for Test",
    "source": "rds",
    "type": "MySQL",
    "database_name": "xxxxx",
    "user_name": "xxxxx",
    "password": "xxxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
Change the data source to GaussDB.
{
    "name": "gaussdb_mysql_instance",
    "description": "Added for Test",
    "source": "gaussdb",
    "type": "MySQL",
    "database_name": "xxxxx",
    "user_name": "xxxxx",
    "password": "xxxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
Change the data source to openGauss.
{
    "name": "gaussdb_instance",
    "description": "Added for Test",
    "source": "gaussdb",
    "type": "OpenGauss",
    "database_name": "xxxxx",
    "user_name": "xxxxx",
    "password": "xxxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
Change the data source to PostgreSQL.
{
    "name": "postgre_instance",
    "description": "Added for Test",
    "source": "rds",
    "type": "PostgreSQL",
    "database_name": "xxxxx",
    "user_name": "xxxxx",
    "password": "xxxxx",
    "region_id": "xxxxx",
    "cluster": {
        "id": "xxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
```


Example Response

Status code: 200

The data sources are successfully updated.

```
{
  "message" : "Update Data Connection Success!"
}
```

Status Codes

Status Code	Description
200	The data sources are successfully updated.

Error Codes

See **Error Codes**.

15.2 Obtaining Data Source Details

Function

This API is used to obtain data source details.

URI

GET /v1/{project_id}/connections/{connection_id}

Table 15-9 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see **Obtaining a Project ID**.
connection_id	Yes	String	Data source ID.

Request Parameters

Table 15-10 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 15-11 Response body parameters

Parameter	Type	Description
id	String	Data source ID.
name	String	Data source name.
description	String	Data source description.
host	String	IP address or domain name of the data source.
port	Integer	Port.
server_list	String	Service list.
database_name	String	Database name.
user_name	String	Username.
url	String	JDBC URL.
test_url	String	JDBC test URL.
password	String	Password for connecting to a database.
source_instance_id	String	ID of the associated cloud instance.
vpc_connection_id	String	VPC connection ID of the associated cloud instance.
project_id	String	Project ID.

Parameter	Type	Description
domain_id	String	User domain ID.
region_id	String	Region ID.
work_space_id	String	Workspace ID.
extra	String	Network configuration information.
config	DcConfigVO object	Configuration information.
type	String	**Explanation** Data source type. For example, **MySQL** indicates that the type of the connected data source is MySQL. The MySQL driver package is used for query. **Constraints** N/A **Value range** The enumerated values are: • **MySQL**. • **DWS**: data source type corresponding to GaussDB(DWS). • **PostgreSQL**. • **OpenGauss**: data source type corresponding to GaussDB. • **Doris**. • **HIVE**. • **ClickHouse**. • **API**: data source type corresponding to API data. **Default value** N/A
source	String	Data source. Enumerated values: • **rds**: RDS data source. • **dws**: GaussDB(DWS) data source. • **mrs**: MRS data source. • **gaussdb**: GaussDB data source. • **public**: public network data source. • **restapi**: API data source. • **endpoint-service**: VPCEP data source.
mode	String	Schema.

Parameter	Type	Description
default_schema	String	Default schema.

Table 15-12 DcConfigVO

Parameter	Type	Description
ssl	Boolean	SSL parameter configuration.
region_id	String	ID of the region where DLI is. Currently, cross-region deployment is not supported. The value must be the ID of the current region.
catalog_name	String	Data directory name. The default value is **dli**. The value of DLI catalogs is **dli**. Default value: **dli**
catalog_type	String	Data directory type. The default value is **dil**. The options include: • **dli**: DLI catalogs. • **lakeformation**: LakeFormation data directories. Default value: **dli**
ds_project_id	String	ID of the project the DLI queue belongs to.
queue_name	String	SQL queue.
auth_type	String	Authentication mode. The value is **AKSK**. Enumerated values: • **AKSK**
ds_ak	String	User AK. Mandatory when the authentication mode is **AKSK**.
ds_sk	String	User SK. Mandatory when the authentication mode is **AKSK**.
time_out	Integer	Request timeout interval when an API is used as the data source.
protocol	String	Protocol. The value is **REST**.
token_key	String	Token key.
token_value	String	Token value.
resp_param	Array of **ResponseParamVO** objects	Response parameters.

Parameter	Type	Description
param_example	String	Parameter example for an API data source.
request_url	String	Request URL when an API is used as the data source.
request_method	String	Request method. The value can be **GET** or **POST**. Enumerated values: • **GET** • **POST**
request_params	Array of **RequestParamDto** objects	List of request parameters.

Table 15-13 ResponseParamVO

Parameter	Type	Description
select	Boolean	Whether the current parameter is selected.
json_path	String	JSON path.
alias_name	String	Parameter alias.
field_type	String	Parameter type. The options include **String**, **Long**, **Decimal**, **DateTime**, and **Bool**.
param_name	String	Parameter name.

Table 15-14 RequestParamDto

Parameter	Type	Description
location	String	Location.
value	Array of strings	Parameter value.
tag_id	String	Tag ID. Mandatory when **param_source** is set to **user**.
param_name	String	Parameter name.
alias_name	String	Parameter alias.

Parameter	Type	Description
field_type	String	Parameter type. The options include **String**, **Long**, **Decimal**, **List<String>**, **List<Long>**, **List<Decimal>**, **DateTime**, **Bool**, **PageIndex**, and **PageNum**.
param_source	String	Parameter source. The options include **custom**, **user**, and **userGroup**.
type	Boolean	Whether to encrypt data.

Example Request

Obtain data source details.

```
GET /v1/{project_id}/connections/{connection_id}
```

Example Response

Status code: 200

Data source details are successfully obtained.

```
{
  "id" : "xxxxxxx3796649a185b012a96xxxxxxx",
  "name" : "xxx_gaussdbtest",
  "host" : "xxx.xx.x.xx",
  "port" : 8000,
  "server_list" : "xxx.xx.x.xx:xxxx",
  "database_name" : "gaussdb",
  "user_name" : "modernbi",
  "test_url" : "jdbc:gaussdb://xxx.xx.xx.xx:xxxx/gaussdb?socketTimeout=120&loginTimeout=5&ssl=true",
  "project_id" : "xxxxxxxb4dac4055888643b33xxxxxxx",
  "domain_id" : "xxxxxxxf3c90428fb22606b64xxxxxxx",
  "work_space_id" : "xxxxxxxea0764ffa850664804xxxxxxx",
  "extra" : "[{\"ip\": \"xxx.xx.x.xx\", \"port\": xxxx, \"channelType\": \"eip\"}]",
  "config" : {
    "ssl" : true
  },
  "type" : "DWS",
  "source" : "dws",
  "mode" : "multi-node",
  "default_schema" : "public",
  "creation_user" : "xxxxxa863a94b0ba2887b80dxxxxx",
  "creation_date" : 1700114987113,
  "creation_user_name" : "test_name",
  "update_user" : "xxxxxxxa863a94b0ba2887b80dcxxxxxxx",
  "update_user_name" : "test_name",
  "update_date" : 1701071948929
}
```

Status Codes

Status Code	Description
200	Data source details are successfully obtained.

Error Codes

See **Error Codes**.

15.3 Deleting a Data Source

Function

This API is used to delete data sources.

URI

DELETE /v1/{project_id}/connections/{connection_id}

Table 15-15 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see **Obtaining a Project ID**.
connection_id	Yes	String	Data source ID.

Request Parameters

Table 15-16 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 15-17 Response body parameter

Parameter	Type	Description
message	String	Message content of the general Message structure.

Example Request

Delete a data source.

```
DELETE /v1/{project_id}/connections/{connection_id}
```

Example Response

Status code: 200

The data source is successfully deleted.

```
{
  "message" : "Delete Data Connection Success!"
}
```

Status Codes

Status Code	Description
200	The data source is successfully deleted.

Error Codes

See **Error Codes**.

15.4 Listing Data Sources

Function

This API is used to list data sources.

URI

GET /v1/{project_id}/connections

Table 15-18 URI parameter

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

Table 15-19 Query parameters

Parameter	Mandatory	Type	Description
all	No	Boolean	Whether to return the list of all data sources.
limit	No	Integer	Maximum number of returned data sources.
offset	No	Integer	Offset.
name	No	String	Data source search value (fuzzy match by name).

Parameter	Mandatory	Type	Description
type	No	String	**Explanation** Data source type. For example, **MySQL** indicates that the type of the connected data source is MySQL. The MySQL driver package is used for query. **Constraints** N/A **Value range** The enumerated values are: • **MySQL**. • **DWS**: data source type corresponding to GaussDB(DWS). • **PostgreSQL**. • **OpenGauss**: data source type corresponding to GaussDB. • **Doris**. • **HIVE**. • **ClickHouse**. • **API**: data source type corresponding to API data. **Default value** N/A
sort_key	No	String	Sorting field. The value can be **name**, **creation_date**, or **update_date**.
sort_dir	No	String	Sorting type. The options are **ASC** and **DESC**. Enumerated values: • **ASC** • **DESC**

Request Parameters

Table 15-20 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 15-21 Response body parameters

Parameter	Type	Description
page_data	Array of **DataConnectionVO** objects	Pagination data. Array length: 0 to 2,147,483,647 elements
count	Long	Total number of data records.

Table 15-22 DataConnectionVO

Parameter	Type	Description
id	String	Data source ID.
name	String	Data source name.
description	String	Data source description.
host	String	IP address or domain name of the data source.
port	Integer	Port.
server_list	String	Service list.
database_name	String	Database name.
user_name	String	Username.

Parameter	Type	Description
url	String	JDBC URL.
test_url	String	JDBC test URL.
password	String	Password for connecting to a database.
source_instance_id	String	ID of the associated cloud instance.
vpc_connection_id	String	VPC connection ID of the associated cloud instance.
project_id	String	Project ID.
domain_id	String	User domain ID.
region_id	String	Region ID.
work_space_id	String	Workspace ID.
extra	String	Network configuration information.
config	DcConfigVO object	Configuration information.
type	String	**Explanation** Data source type. For example, **MySQL** indicates that the type of the connected data source is MySQL. The MySQL driver package is used for query. **Constraints** N/A **Value range** The enumerated values are: • **MySQL**. • **DWS**: data source type corresponding to GaussDB(DWS). • **PostgreSQL**. • **OpenGauss**: data source type corresponding to GaussDB. • **Doris**. • **HIVE**. • **ClickHouse**. • **API**: data source type corresponding to API data. **Default value** N/A

Parameter	Type	Description
source	String	Data source. Enumerated values: • **rds**: RDS data source. • **dws**: GaussDB(DWS) data source. • **mrs**: MRS data source. • **gaussdb**: GaussDB data source. • **public**: public network data source. • **restapi**: API data source. • **endpoint-service**: VPCEP data source.
mode	String	Schema.
default_schema	String	Default schema.

Table 15-23 DcConfigVO

Parameter	Type	Description
ssl	Boolean	SSL parameter configuration.
region_id	String	Region ID. Currently, cross-region deployment is not supported. The value must be the ID of the current region.
catalog_name	String	Data directory name. The default value is **dli**. The value of DLI catalogs is **dli**.
catalog_type	String	Data directory type. The default value is **dil**. The options include: • **dli**: DLI catalogs. • **lakeformation**: LakeFormation data directories.
ds_project_id	String	ID of the project the DLI queue belongs to.
queue_name	String	SQL queue.
auth_type	String	Authentication mode. The value is **AKSK**. Enumerated values: • **AKSK**
ds_ak	String	User AK. Mandatory when the authentication mode is **AKSK**.
ds_sk	String	User SK. Mandatory when the authentication mode is **AKSK**.
time_out	Integer	Request timeout interval when an API is used as the data source.

Parameter	Type	Description
protocol	String	Protocol. The value is **REST**.
token_key	String	Token key.
token_value	String	Token value.
resp_param	Array of **ResponseParamVO** objects	Response parameters.
param_example	String	Parameter example for an API data source.
request_url	String	Request URL when an API is used as the data source.
request_method	String	Request method. The value can be **GET** or **POST**. Enumerated values: • **GET** • **POST**
request_params	Array of **RequestParamDto** objects	List of request parameters.

Table 15-24 ResponseParamVO

Parameter	Type	Description
select	Boolean	Whether the current parameter is selected.
json_path	String	JSON path.
alias_name	String	Parameter alias.
field_type	String	Parameter type. The options include **String**, **Long**, **Decimal**, **DateTime**, and **Bool**.
param_name	String	Parameter name.

Table 15-25 RequestParamDto

Parameter	Type	Description
location	String	Location.
value	Array of strings	Parameter value.

Parameter	Type	Description
tag_id	String	Tag ID. Mandatory when **param_source** is set to **user**.
param_name	String	Parameter name.
alias_name	String	Parameter alias.
field_type	String	Parameter type. The options include **String**, **Long**, **Decimal**, **List<String>**, **List<Long>**, **List<Decimal>**, **DateTime**, **Bool**, **PageIndex**, and **PageNum**.
param_source	String	Parameter source. The options include **custom**, **user**, and **userGroup**.
type	Boolean	Whether to encrypt data.

Example Request

List data sources.

```
GET /v1/{project_id}/connections
```

Example Response

Status code: 200

Data sources are successfully listed.

```
{
  "count" : 100,
  "page_data" : [ {
    "id" : "xxxxxxx796649a185b012a96xxxxxxx",
    "name" : "test_name",
    "host" : "xxx.xx.x.xx",
    "port" : 80,
    "server_list" : "xxx.xx.x.xx:xxxx",
    "database_name" : "db_name",
    "user_name" : "test_name",
    "test_url" : "jdbc:gaussdb://xxx.xx.x.xx:xxxx/gaussdb?socketTimeout=120&loginTimeout=5&ssl=true",
    "project_id" : "xxxxxxxb4dac4055888643b331axxxxxxx",
    "domain_id" : "xxxxxxxcf3c90428fb22606b64dxxxxxxx",
    "work_space_id" : "xxxxxxxa5ea0764ffa850664804b9xxxxxxx",
    "extra" : "[{\"ip\": \"xxx.xx.x.xx\", \"port\": xxxx, \"channelType\": \"eip\"}]",
    "config" : {
      "ssl" : true
    },
    "type" : "DWS",
    "source" : "dws",
    "mode" : "multi-node",
    "default_schema" : "public",
    "creation_user" : "xxxxxxx63a94b0ba2887b8xxxxxxx",
    "creation_date" : 1700114987113,
    "creation_user_name" : "ei_dlv_l00456193_01",
    "update_user" : "xxxxxxxa863a94b0ba2887b80dxxxxxxx",
    "update_user_name" : "name1",
    "update_date" : 1701071948929
  } ]
}
```


Status Codes

Status Code	Description
200	Data sources are successfully listed.

Error Codes

See **Error Codes**.

15.5 Creating a Data Source

Function

This API is used to create data sources.

URI

POST /v1/{project_id}/connections

Table 15-26 URI parameter

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

Request Parameters

Table 15-27 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 15-28 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Data source name.
description	No	String	Data source description.
type	Yes	String	**Explanation** Data source type. For example, **MySQL** indicates that the type of the connected data source is MySQL. The MySQL driver package is used for query. **Constraints** N/A **Value range** The enumerated values are: • **MySQL**. • **DWS**: data source type corresponding to GaussDB(DWS). • **PostgreSQL**. • **OpenGauss**: data source type corresponding to GaussDB. • **Doris**. • **HIVE**. • **ClickHouse**. • **API**: data source type corresponding to API data. **Default value** N/A

Parameter	Mandatory	Type	Description
source	Yes	String	Data source. Enumerated values: • **rds**: RDS data source. • **dws**: GaussDB(DWS) data source. • **mrs**: MRS data source. • **gaussdb**: GaussDB data source. • **public**: public network data source. • **restapi**: API data source. • **endpoint-service**: VPCEP data source.
server_list	No	String	List of servers. Mandatory if the data source is in cluster mode (containing multiple instances), for example, **DWS** or **ClickHouse**.
database_name	Yes	String	Database name.
host	No	String	Domain name. Mandatory for non-cluster data sources, for example, **mysql**.
port	No	Integer	Port.
user_name	No	String	Username.
password	No	String	Password.
region_id	No	String	Current region ID.
cluster	No	**ClusterInfoDTO** object	Mandatory when any cloud instance is connected.
config	Yes	**DcConfigDTO** object	Configuration parameter.

Table 15-29 ClusterInfoDTO

Parameter	Mandatory	Type	Description
id	No	String	ID of the associated cloud instance.

Table 15-30 DcConfigDTO

Parameter	Mandatory	Type	Description
ssl	No	Boolean	SSL parameter configuration.
region_id	No	String	ID of the region where DLI is. Currently, cross-region deployment is not supported. The value must be the ID of the current region.
catalog_name	No	String	Data directory name. The default value is **dli**. The value of DLI catalogs is **dli**. Default value: **dli**
catalog_type	No	String	Data directory type. The default value is **dil**. The options include: • **dli**: DLI catalogs. • **lakeformation**: LakeFormation data directories. Default value: **dli**
ds_project_id	No	String	ID of the project the DLI queue belongs to.
queue_name	No	String	SQL queue.
auth_type	No	String	Authentication mode. The value is **AKSK**. Enumerated values: • **AKSK**
ds_ak	No	String	User AK. Mandatory when the authentication mode is **AKSK**.
ds_sk	No	String	User SK. Mandatory when the authentication mode is **AKSK**.
time_out	No	Integer	Request timeout interval when an API is used as the data source.
protocol	No	String	Protocol. The value is **REST**.
token_key	No	String	Token key.
token_value	No	String	Token value.
request_url	No	String	Request URL when an API is used as the data source.

Parameter	Mandatory	Type	Description
request_method	No	String	Request method. The value can be **GET** or **POST**. Enumerated values: • **GET** • **POST**
request_params	No	Array of **RequestParamDto** objects	List of request parameters.
resp_param	No	Array of **ResponseParamDto** objects	Response parameters.

Table 15-31 RequestParamDto

Parameter	Mandatory	Type	Description
location	Yes	String	Location.
value	Yes	Array of strings	Parameter value.
tag_id	No	String	Tag ID. Mandatory when **param_source** is set to **user**.
param_name	Yes	String	Parameter name.
alias_name	No	String	Parameter alias.
field_type	Yes	String	Parameter type. The options include **String**, **Long**, **Decimal**, **List<String>**, **List<Long>**, **List<Decimal>**, **DateTime**, **Bool**, **PageIndex**, and **PageNum**.
param_source	Yes	String	Parameter source. The options include **custom**, **user**, and **userGroup**.
type	Yes	Boolean	Whether to encrypt data.

Table 15-32 ResponseParamDto

Parameter	Mandatory	Type	Description
alias_name	Yes	String	Parameter alias.

Parameter	Mandatory	Type	Description
field_type	Yes	String	Parameter type. The options include **String**, **Long**, **Decimal**, **DateTime**, and **Bool**.
param_name	Yes	String	Parameter name.

Response Parameters

Status code: 200

Table 15-33 Response body parameter

Parameter	Type	Description
message	String	Description.

Example Request

Create data sources.

```
POST /v1/{project_id}/connections
Create a public network data source.
{
  "name": "dws02",
  "server_list": "xx.xx.x.xx:xxxx",
  "database_name": "gaussdb",
  "user_name": "xxxxx",
  "password": "xxxxxxx",
  "config": {
    "ssl": true
  },
  "source": "public",
  "type": "DWS"
}
Create a GaussDB(DWS) data source.
{
    "name": "dws04_instance",
    "description": "Added for Test",
    "source": "dws",
    "type": "DWS",
    "database_name": "xxxxx",
    "user_name": "xxxx",
    "password": "xxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
Create an RDS data source.
{
    "name": "mysql_instance",
    "description": "Added for Test",
    "source": "rds",
    "type": "MySQL",
```



```
    "database_name": "xxxxx",
    "user_name": "xxxxx",
    "password": "xxxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
Create a GaussDB data source.
{
    "name": "gaussdb_mysql_instance",
    "description": "Added for Test",
    "source": "gaussdb",
    "type": "MySQL",
    "database_name": "xxxxx",
    "user_name": "xxxxx",
    "password": "xxxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
Create an openGauss data source.
{
    "name": "gaussdb_instance",
    "description": "Added for Test",
    "source": "gaussdb",
    "type": "OpenGauss",
    "database_name": "xxxxx",
    "user_name": "xxxxx",
    "password": "xxxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
Create a PostgreSQL data source.
{
    "name": "postgre_instance",
    "description": "Added for Test",
    "source": "rds",
    "type": "PostgreSQL",
    "database_name": "xxxxx",
    "user_name": "xxxxx",
    "password": "xxxxx",
    "region_id": "xxxx",
    "cluster": {
        "id": "xxxxxxxx"
    },
    "config": {
        "ssl": false
    }
}
```

Example Response

Status code: 200

Data sources are successfully created.


```
{
  "message" : "xxxxxxxeac2af4e738e2430bb44xxxxxxx"
}
```

Status Codes

Status Code	Description
200	Data sources are successfully created.

Error Codes

See **Error Codes**.

15.6 Verifying a VPC Endpoint Service and Creating a VPC Endpoint

Function

This API is used to verify VPC endpoint services and create VPC endpoints.

URI

POST /v1/{project_id}/connections/vpc-endpoint

Table 15-34 URI parameter

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

Request Parameters

Table 15-35 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 15-36 Request body parameters

Parameter	Mandatory	Type	Description
ds_type	Yes	String	**Explanation** Data source type. For example, **MySQL** indicates that the type of the connected data source is MySQL. The MySQL driver package is used for query. **Constraints** N/A **Value range** The enumerated values are: • **MySQL**. • **DWS**: data source type corresponding to GaussDB(DWS). • **PostgreSQL**. • **OpenGauss**: data source type corresponding to GaussDB. • **Doris**. • **HIVE**. • **ClickHouse**. • **API**: data source type corresponding to API data. **Default value** N/A
vpc_ep_server_name	No	String	Name of a VPC endpoint service. This parameter is mandatory when the VPC endpoint service ID is left blank.

Parameter	Mandatory	Type	Description
vpc_ep_server_id	No	String	ID of the VPC endpoint service. This parameter is mandatory when the VPC endpoint service name is left blank.

Response Parameters

Status code: 200

Table 15-37 Response body parameters

Parameter	Type	Description
id	String	ID of the connected instance.
create_date	Long	Timestamp when the connected instance is created.
create_user	String	Creator ID.
create_user_name	String	Creator name.
update_date	Integer	Timestamp when the connected instance is updated.
update_user	String	ID of the user who performs the update.
update_user_name	String	Name of the user who performs the update.
project_id	String	Project ID.
domain_id	String	User domain ID.

Parameter	Type	Description
ds_type	String	**Explanation** Data source type. For example, **MySQL** indicates that the type of the connected data source is MySQL. The MySQL driver package is used for query. **Constraints** N/A **Value range** The enumerated values are: • **MySQL**. • **DWS**. • **PostgreSQL**. • **OpenGauss**: data source type corresponding to GaussDB. • **Doris**. • **HIVE**. • **ClickHouse**. • **API**: data source type corresponding to API data. **Default value** N/A
vpc_ep_server_id	String	ID of the VPC endpoint service.
vpc_ep_server_name	String	Name of a VPC endpoint service.
vpc_ep_client_id	String	VPC endpoint ID.
vpc_ep_client_status	String	VPC endpoint status. The options are: • **accepted**: The VPC endpoint has been accepted. • **pendingAcceptance**: The VPC endpoint is to be reviewed.
scene	String	Network scenario. • **dii-app_to_data-source**: VPCEP channel from DataArts Insight to the data source.

Example Request

Verify a VPC endpoint service and create a VPC endpoint.


```
POST /v1/{project_id}/connections/vpc-endpoint
{
  "ds_type" : "DWS",
  "vpc_ep_server_name" : "cn-xxxx-x.DII-INGEST-LB.xxxxxx-31cf-4e27-xxxx-4b71db9b82fb"
}
```

Example Response

Status code: 200

The VPC endpoint service is successfully verified.

```
{
  "create_date" : 1714962072863,
  "create_user" : "xxxxx",
  "create_user_name" : "xxxx",
  "update_date" : 1714962072863,
  "update_user" : "xxxx",
  "update_user_name" : "xxxx",
  "id" : "8a8080c08xxxxxf4bb485240001",
  "project_id" : "9c3043axxxx88643b3xxxxxx27",
  "domain_id" : "d22f8axxx428fb2260xxxxx823",
  "ds_type" : "DWS",
  "vpc_ep_server_id" : "xxxxx-31cf-4e27-xxxxx-4b71db9b82fb",
  "vpc_ep_server_name" : "cn-xxxx-x.DII-xxxx-LB.716b4ee7-xxxx-4e27-852e-xxxxxxx",
  "vpc_ep_client_id" : "xxxxxx-62be-4479-xxxxx-7379d6dafc40",
  "vpc_ep_client_status" : "accepted",
  "scene" : "dii-app_to_data-source"
}
```

Status Codes

Status Code	Description
200	The VPC endpoint service is successfully verified.

Error Codes

See **Error Codes**.

16 Pass Management

16.1 Listing Allocated, Inactive, and All IAM Users

Function

This API is used to list allocated, inactive, and all IAM users.

URI

GET /v1/{project_id}/quota-users/all

Table 16-1 URI parameter

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

Table 16-2 Query parameters

Parameter	Mandatory	Type	Description
account_name	No	String	Username (fuzzy search).
sys_role_list	No	Array	System-defined roles. The options include: • **0**: inactive user. • **1**: general user. • **2**: read-only user.

Request Parameters

Table 16-3 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 16-4 Response body parameter

Parameter	Type	Description
[Array element]	Array of **Table 16-5** objects	-

Table 16-5 UserQuotaVo

Parameter	Type	Description
user_id	String	User ID.
account_name	String	Username.
user_type	String	User type. The options are **IAM**, **SELF-BUILT**, and **DEFAULT**.
sys_role	Integer	User role. The options include: • **0**: inactive user. • **1**: general user. • **2**: read-only user.
effective_time	Long	Effective timestamp.

Example Request

```
GET https://{endpoint}/v1/{project_id}/quota-users/all?account_name=&sys_role_list=
```


Example Response

None

Status Codes

Status Code	Description
200	All users are successfully listed.

Error Codes

See **Error Codes**.

16.2 Updating the Status of a User Pass

Function

This API is used to update statuses of user passes.

URI

PUT /v1/{project_id}/quota-users

Table 16-6 URI parameter

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

Request Parameters

Table 16-7 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 16-8 Request body parameters

Parameter	Mandatory	Type	Description
user_id_list	Yes	Array of strings	User ID list.
sys_role	Yes	Integer	User role. The options include: • **0**: inactive user. • **1**: general user. • **2**: read-only user.

Response Parameters

Status code: 200

Table 16-9 Response body parameter

Parameter	Type	Description
data	Boolean	Whether the request is successful. The options include **true** and **false**.

Example Request

```
PUT https://{endpoint}/v1/{project_id}/quota-users
```

Example Response

None

Status Codes

Status Code	Description
200	The user pass status is successfully edited.

Error Codes

See **Error Codes**.

16.3 Adding Quota-Associated Users for Scenarios Such as Federated Login

Function

This API is used to add quota-associated users for scenarios such as federated login.

URI

POST /v1/{project_id}/quota-users

Table 16-10 URI parameter

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

Request Parameters

Table 16-11 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 16-12 Request body parameters

Parameter	Mandatory	Type	Description
user_name_list	Yes	Array of strings	List of user names.
sys_role	Yes	Integer	Pass role. The options include: • **0**: inactive user. • **1**: general user. • **2**: read-only user.

Response Parameters

Status code: 200

Example Request

POST https://{endpoint}/v1/{project_id}/quota-users

Example Response

None

Status Codes

Status Code	Description
200	Users are successfully added.

Error Codes

See **Error Codes**.

16.4 Listing the Users Who Have Been Assigned Passes (Excluding Inactive Users)

Function

This API is used to list the users who have been assigned passes (excluding inactive users).

URI

GET /v1/{project_id}/quota-users

Table 16-13 URI parameter

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

Table 16-14 Query parameters

Parameter	Mandatory	Type	Description
account_name	No	String	Username (fuzzy search).
sys_role	No	String	System-defined roles. The options include: • **1**: general user. • **2**: read-only user.
type	No	String	Role type. The value can be **IAM** or **SELF-BUILT**.
sort_key	No	String	Sorting field. The options include **effectiveTime** and **accountName**.
sort_dir	No	String	Sorting mode. The options are **desc** and **asc**.
offset	No	Integer	Number of pages. The default value is **0**.
limit	No	Integer	Number of data records on each page. The default value is **10**.

Request Parameters

Table 16-15 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 16-16 Response body parameters

Parameter	Type	Description
count	Integer	Total data volume.
page_data	Array of **Table 16-17** objects	Data.

Table 16-17 page_data

Parameter	Type	Description
user_id	String	User ID.
account_name	String	Username.
user_type	String	User type. The options are **IAM**, **SELF-BUILT**, and **DEFAULT**.
sys_role	Integer	User role. The options include: • **0**: inactive user. • **1**: general user. • **2**: read-only user. Enumerated values: • **0** • **1** • **2**
effective_time	Long	Effective timestamp.

Example Request

```
GET https://{endpoint}/v1/{project_id}/quota-users?
sort_key=accountName&sort_dir=asc&limit=10&offset=0&account_name=&sys_role=1
```

Example Response

None

Status Codes

Status Code	Description
200	Passes are successfully listed.

Error Codes

See **Error Codes**.

16.5 Deleting User Allocation Information

Function

This API is used to delete user allocation information.

URI

DELETE /v1/{project_id}/quota-users/{user_id}

Table 16-18 URI parameter

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

Request Parameters

Table 16-19 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 16-20 Response body parameter

Parameter	Type	Description
data	Boolean	Whether the request is successful. The options include **true** and **false**.

Example Request

```
DELETE https://{endpoint}/v1/{project_id}/quota-users/{user_id}
```

Example Response

None

Status Codes

Status Code	Description
200	The user is successfully deleted.

Error Codes

See **Error Codes**.

17 Dashboard

17.1 Creating a Dashboard

Function

This API is used to create dashboards.

URI

POST /v1/{project_id}/dashboards/create

Table 17-1 URI parameter

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

Request Parameters

Table 17-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 17-3 Request body parameter

Parameter	Mandatory	Type	Description
name	Yes	String	Dashboard name.

Response Parameters

Status code: 200

Table 17-4 Response body parameters

Parameter	Type	Description
id	String	Dashboard ID.
name	String	Dashboard name.
create_user_name	String	Creator.
create_date	Long	Creation time.
update_user_name	String	User who performs the update.
update_date	Long	Update timestamp.
status	Integer	Dashboard status. The options include: • **0**: not published. • **1**: published. • **2**: taken offline. Enumerated values: • **0** • **1** • **2**

Example Request

Create a dashboard.

```
https://{endpoint}/v1/{project_id}/dashboards/create
{
  "name" : "newDashboard"
}
```

Example Response

Status code: 200

The dashboard is successfully created.

```
{
  "id" : "009b3d3e-xxxx-4cc3-xxxx-dc54cd42424d",
  "name": "New Dashboard",
  "create_user_name" : "xxxx",
  "create_date" : 1706062051243,
  "update_user_name" : "xxxx",
  "update_date" : 1706062051243,
  "status" : 0
}
```

Status Codes

Status Code	Description
200	The dashboard is successfully created.

Error Codes

See **Error Codes**.

17.2 Updating a Dashboard

Function

This API is used to update dashboards.

URI

POST /v1/{project_id}/dashboards/update

Table 17-5 URI parameter

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

Request Parameters

Table 17-6 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 17-7 Request body parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Dashboard ID.
name	Yes	String	Dashboard name.

Response Parameters

Status code: 200

Table 17-8 Response body parameters

Parameter	Type	Description
id	String	Dashboard ID.
name	String	Dashboard name.
create_user_name	String	Creator.
create_date	Long	Creation timestamp.
update_user_name	String	User who performs the update.
update_date	Long	Update timestamp.

Parameter	Type	Description
status	Integer	Dashboard status. The options include: • **0**: not published. • **1**: published. • **2**: taken offline. Enumerated values: • **0** • **1** • **2**

Example Request

Update a dashboard.

```
https://{endpoint}/v1/{project_id}/dashboards/update
{
  "id" : "009b3d3e-xxxx-4cc3-xxxx-dc54cd42424d",
  "name": "Update Dashboard"
}
```

Example Response

Status code: 200

The dashboard is successfully updated.

```
{
  "id" : "009b3d3e-xxxx-4cc3-xxxx-dc54cd42424d",
  "name": "Update Dashboard",
  "create_user_name" : "xxxx",
  "create_date" : 1706062051243,
  "update_user_name" : "xxxx",
  "update_date" : 1706062051243,
  "status" : 0
}
```

Status Codes

Status Code	Description
200	The dashboard is successfully updated.

Error Codes

See **Error Codes**.

17.3 Batch Publishing Dashboards

Function

This API is used to publish dashboards in batches.

URI

POST /v1/{project_id}/dashboards/batch-publish

Table 17-9 URI parameter

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

Request Parameters

Table 17-10 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 17-11 Request body parameters

Parameter	Mandatory	Type	Description
ids	Yes	Array of strings	List of IDs of dashboards to be batch published.

Response Parameters

Status code: 200

Table 17-12 Response body parameter

Parameter	Type	Description
data	Boolean	Whether the request is successful. The options include **true** and **false**.

Example Request

Publish dashboards in batches.

```
https://{endpoint}/v1/{project_id}/dashboards/batch-publish
{
  "ids" : [ "009b3d3e-xxxx-4cc3-xxxx-dc54cd42xxxx" ]
}
```

Example Response

Status code: 200

Dashboards are successfully published in batches.

```
{
  "data" : true
}
```

Status Codes

Status Code	Description
200	Dashboards are successfully published in batches.

Error Codes

See **Error Codes**.

17.4 Batch Taking Dashboards Offline

Function

This API is used to take dashboards offline in batches.

URI

POST /v1/{project_id}/dashboards/batch-offline

Table 17-13 URI parameter

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

Request Parameters

Table 17-14 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 17-15 Request body parameter

Parameter	Mandatory	Type	Description
ids	Yes	Array of strings	List of IDs of dashboards to be batch taken offline.

Response Parameters

Status code: 200

Table 17-16 Response body parameter

Parameter	Type	Description
data	Boolean	Whether the request is successful. The options include **true** and **false**.

Example Request

Take dashboards offline in batches.


```
https://{endpoint}/v1/{project_id}/dashboards/batch-offline
{
  "ids" : [ "009b3d3e-xxxx-4cc3-xxxx-dc54cd42xxxx" ]
}
```

Example Response

Status code: 200

Dashboards are successfully taken offline in batches.

```
{
  "data" : true
}
```

Status Codes

Status Code	Description
200	Dashboards are successfully taken offline in batches.

Error Codes

See **Error Codes**.

17.5 Batch Deleting Dashboards

Function

This API is used to delete dashboards in batches.

URI

POST /v1/{project_id}/dashboards/batch-delete

Table 17-17 URI parameter

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

Request Parameters

Table 17-18 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 17-19 Request body parameters

Parameter	Mandatory	Type	Description
ids	Yes	Array of strings	List of IDs of dashboards to be batch deleted.

Response Parameters

Status code: 200

Table 17-20 Response body parameter

Parameter	Type	Description
data	Boolean	Whether the request is successful. The options include **true** and **false**.

Example Request

Delete dashboards in batches.

```
https://{endpoint}/v1/{project_id}/dashboards/batch-delete
{
  "ids" : [ "009b3d3e-xxxx-4cc3-xxxx-dc54cd42xxxx" ]
}
```

Example Response

Status code: 200

Dashboards are successfully deleted in batches.


```
{
  "data" : true
}
```

Status Codes

Status Code	Description
200	Dashboards are successfully deleted in batches.

Error Codes

See **Error Codes**.

18 Large Screen

18.1 Listing Large Screen Components

Function

This API is used to list large screen components.

URI

GET /v1/{project_id}/screens/{screen_id}/nodes

Table 18-1 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see **Obtaining a Project ID**.
screen_id	Yes	String	Large screen ID.

Table 18-2 Query parameters

Parameter	Mandatory	Type	Description
name	No	String	Component name (fuzzy search).

Parameter	Mandatory	Type	Description
type	No	String	Component type. Enumerated values: • **line**: line chart. • **type**: combination chart. • **linebardoubley**: dual Y-axis combination chart. • **linestep**: step line chart. • **pie**: pie chart. • **pie_percent**: circular progress bar. • **rosepie**: rose chart. • **bar**: bar chart. • **barhori**: horizontal bar chart. • **linebar**: combination chart. • **linebardoubley**: dual Y-axis combination chart. • **gantt**: gantt chart. • **map**: China map. • **worldmap**: world map. • **custommap**: custom map. • **img**: image. • **video**: video. • **title**: title. • **text**: text. • **wcloud**: word cloud. • **datetime**: timer. • **table**: carousel table. • **tablepage**: table. • **flask**: ticker board. • **thresholdflask**: threshold board. • **marquee**: marquee. • **broadcastlist**: carousel bar chart. • **treemap**: treemap chart. • **border**: border. • **decorate**: decoration. • **bgcom**: background image. • **fullscreen**: full screen.

Parameter	Mandatory	Type	Description
			• **iframe**: iFrame. • **search**: search. • **select**: single-select dropdown list. • **tab**: tab list. • **mutiselect**: multi-select dropdown list. • **dates**: date selector. • **dashboard**: dashboard. • **funnel**: funnel chart. • **tree**: tree chart. • **progress**: progress bar. • **distributionmap**: industry distribution chart. • **milestone**: milestone.
has_data_bind	No	Boolean	Whether data is bound to the component.

Request Parameters

Table 18-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 18-4 Response body parameters

Parameter	Type	Description
id	String	Large screen ID
name	String	Large screen name.
pages	Array of **ScreenPageOpenApiVO** objects	Large screen page list.

Table 18-5 ScreenPageOpenApiVO

Parameter	Type	Description
id	String	Page ID.
name	String	Page name.
nodes	Array of **ScreenNodeOpenApiVO** objects	List of components on a page.

Table 18-6 ScreenNodeOpenApiVO

Parameter	Type	Description
id	String	Component ID.
name	String	Component name.
type	String	Component type.
hidden	Boolean	Whether the component is hidden.
target_nodes	Array of **target_nodes** objects	Component associated with an interactive event.

Table 18-7 target_nodes

Parameter	Type	Description
id	String	ID of an associated component.
field_id	String	Field ID of the associated component.

Example Request

List large screen components.

```
GET https://{endpoint}/v1/{project_id}/screens/{screen_id}/nodes?
name=node1&type=line&has_data_bind=true
```

Example Response

Status code: 200

Large screen components are successfully listed.

```
{
  "id" : "b7401134-xxxx-4973-xxxx-2b6c1ac3f2f1",
  "name" : "test_screen",
  "pages" : [ {
    "id" : "ae613423-xxxx-475d-xxxx-be6fc28c3d30",
    "name" : null,
    "nodes" : [ {
      "id" : "d1e4b677-xxxx-475d-xxxx-dbd9a4bf2f20",
      "name" : "Single-Select Dropdown List 1",
      "type" : "select",
      "hidden" : false,
      "target_nodes" : [ {
        "id" : "9f7ff361-xxxx-475d-xxxx-57dd19bda14e",
        "field_id" : "cca5cb5e-xxxx-475d-xxxx-8a4f90447ae0.field"
      } ]
    } ]
  } ]
}
```

Status Codes

Status Code	Description
200	Large screen components are successfully listed.

Error Codes

See **Error Codes**.

18.2 Retrieving the Data of a Specified Large Screen Component

Function

This API is used to retrieve the data of a specified large screen component.

URI

POST /v1/{project_id}/screens/{screen_id}/query-data

Table 18-8 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see **Obtaining a Project ID**.
screen_id	Yes	String	Large screen ID.

Request Parameters

Table 18-9 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Table 18-10 Request body parameters

Parameter	Mandatory	Type	Description
node_id	Yes	String	Component ID.
selectors	No	Array of **selectors** objects	Filter list.

Table 18-11 selectors

Parameter	Mandatory	Type	Description
selector_node_id	No	String	Filter component ID.
field_id	No	String	Filter field ID.
values	No	Array of objects	Filter value.

Response Parameters

Status code: 200

Table 18-12 Response body parameters

Parameter	Type	Description
cell_data	Array of **OpenApiCellInfoList** objects	List of returned data. By default, the first line is the field name.
record_count	Integer	Number of returned data records.

Table 18-13 OpenApiCellInfoList

Parameter	Type	Description
cell_info_list	Array of **OpenApiCellInfo** objects	Data list.

Table 18-14 OpenApiCellInfo

Parameter	Type	Description
caption	String	Field name.
data_type	String	Data type. The options include: • **DATETIME**: time type. • **STRING**: character type. • **NUMBER**: numeric type.
level_type	String	Subtype of a date, for example, year, month, and day.
cell_raw_value	Object	Original value of a field.
cell_value	Object	Formatted value of a field.
model_type	String	Field type. The options are: • **dimension**: dimension. • **measure**: metric. • **metric**: indicator.

Example Request

Retrieve the data of a specified large screen component.

```
https://{endpoint}/v1/{project_id}/screens/{screen_id}/query-data

{
  "node_id" : "9f7ff361-xxxx-482d-xxxx-57dd19bda14e",
  "selectors" : [ {
    "selector_node_id" : "d1e4b677-xxxx-4016-xxxx-dbd9a4bf2f20",
    "field_id" : "cca5cb5e-xxxx-4614-xxxx-8a4f90447ae0.field",
    "values" : [ "value1" ]
  } ]
}
```

Example Response

Status code: 200

The component data is successfully returned.

```
{
  "cell_data" : [ [ {
    "caption" : "create_time(year)",
    "data_type" : "DATETIME",
    "level_type" : "yearLevel",
    "cell_raw_value" : null,
    "cell_value" : "create_time(year)",
    "model_type" : "dimension"
  } ], [ {
    "caption" : "create_time(year)",
    "data_type" : "DATETIME",
    "level_type" : "yearLevel",
    "cell_raw_value" : "2020",
    "cell_value" : "2020",
    "model_type" : "dimension"
  } ] ],
  "record_count" : 1
}
```

Status Codes

Status Code	Description
200	The component data is successfully returned.

Error Codes

See **Error Codes**.

19 Intelligent Analysis Assistant

19.1 Listing Intelligent Analysis Assistants

Function

This API is used to list intelligent analysis assistants.

URI

GET /v1/{project_id}/subjects

Table 19-1 URI parameter

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

Table 19-2 Query parameters

Parameter	Mandatory	Type	Description
keyword	No	String	Name of the intelligent analysis assistant.
sort_dir	No	String	Sorting type. The options include: • **desc**: descending order. • **asc**: ascending order. Enumerated values: • **desc** • **asc**

Parameter	Mandatory	Type	Description
sort_key	No	String	Sorting field.
limit	No	Integer	Number of returned entries. If not transferred, the first 10 records are queried by default.
offset	No	Integer	Start offset of returned entries.

Request Parameters

Table 19-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.

Response Parameters

Status code: 200

Table 19-4 Response body parameters

Parameter	Type	Description
count	Integer	Total number of entries.
page_data	Array of **ExtSubjectVO** objects	Data on the current page. The element type is **ExtSubjectVO**.

Table 19-5 ExtSubjectVO

Parameter	Type	Description
name	String	Name of the intelligent analysis assistant.

Parameter	Type	Description
id	String	ID of the intelligent analysis assistant.
comments	String	Description of the intelligent analysis assistant.
status	Integer	Status of the intelligent analysis assistant. The options include **0** (initialization), **1** (chart construction), **2** (index construction), **3** (successful training), **4** (failed training), and **5** (stopped training). Enumerated values: • **0** • **1** • **2** • **3** • **4** • **5**
permission_list	Array of strings	Current user's sharing and collaboration permissions. The options are **edit**, **export**, **use**, and **read**.
workspace_id	String	ID of the workspace where the intelligent analysis assistant is located.
project_id	String	ID of the project the intelligent analysis assistant belongs to.
create_user	String	Creator of the intelligent analysis assistant.
create_user_name	String	Name of the creator of the intelligent analysis assistant.
create_date	Long	Time when the intelligent analysis assistant was created.
update_user	String	User who updates the intelligent analysis assistant.
update_user_name	String	Name of the user who updates the intelligent analysis assistant.
update_date	Long	Time when the intelligent analysis assistant was updated.
datasets	Array of **ExtSubjectDatasetVO** objects	List of datasets associated with the intelligent analysis assistant.
recommended_questions	String	Questions recommended by the intelligent analysis assistant.

Table 19-6 ExtSubjectDatasetVO

Parameter	Type	Description
dataset_id	String	Dataset ID.
name	String	Dataset name.

Example Request

List intelligent analysis assistants.

```
GET https://{endpoint}/v1/{project_id}/subjects?subjects?keyword=&sort_key=&sort_dir=&limit=20&offset=0
```

Example Response

Status code: 200

Intelligent analysis assistants are successfully listed.


```
{
  "count" : 1,
  "page_data" : [ {
    "name" : "xxxxxxxxx",
    "subject_id" : "xxxxxxxxxxxxx",
    "comments" : "",
    "status" : 3,
    "permission_list" : [ "edit", "use" ],
    "workspace_id" : "xx4xxaxxxxac8x51xxx30",
    "project_id" : "xxb4dxx4xxx886xx33xx27",
    "create_user" : "xxxb4xxx4b0baxxxxx0dc18xxx3",
    "create_user_name" : "xxv_l00xxxx",
    "create_date" : 1702046652513,
    "update_user" : "xxxxxxx4b0xxxb8xxxx27x",
    "update_user_name" : "xxv_l00xxxx",
    "update_date" : 1710496061316,
    "datasets" : [ {
      "dataset_id" : "xxxxxxxxxxxxx",
      "name" : "xxxxxxxxxx"
    } ],
    "recommended_questions" : "xxxxxxxxxxxxx"
  } ]
}
```


Status Codes

Status Code	Description
200	Intelligent analysis assistants are successfully listed.

Error Codes

See **Error Codes**.

19.2 Asking Data Questions in Natural Language

Function

This is a natural language query API that allows you to input data query questions in natural language and receive the corresponding SQL and data as output.

URI

POST /v1/{project_id}/nl-query

Table 19-7 URI parameter

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

Request Parameters

Table 19-8 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Obtain a user token by calling IAM's "Obtaining a User Token" API. **X-Subject-Token** in the response header is the desired user token.
X-Workspace-Id	Yes	String	Workspace ID. For how to obtain it, see **Obtaining a Workspace ID**.

Table 19-9 Request body parameters

Parameter	Mandatory	Type	Description
subject_id	Yes	String	ID of the intelligent analysis assistant.

Parameter	Mandatory	Type	Description
dialog_id	Yes	String	Dialog ID. If you do not want to trigger multi-turn Q&A, you need to pass in a different dialog ID each time you make a request.
message_content	Yes	String	Question.
need_insight	No	Boolean	Whether intelligent insights are required.

Response Parameters

Status code: 200

Table 19-10 Response body parameters

Parameter	Type	Description
query_data	Array<Array<QueryCellInfo>>	Data queried by the user.
sql_text	String	SQL statements generated by the query.
insight	String	Intelligent insights generated by the query.

Table 19-11 QueryCellInfo

Parameter	Type	Description
caption	String	Attribute name, which is sourced from the dataset configuration information.
data_type	String	Attribute type. Enumerated values: • **DATE** • **NUMBER** • **STRING**

Parameter	Type	Description
level_type	String	Date type. Enumerated values: • **yearLevel** • **quarterLevel** • **monthLevel** • **weekLevel** • **dayLevel** • **hourLevel** • **minuteLevel** • **secondLevel** • **ymdhmsLevel**
cell_raw_value	Object	Original value of a cell.
cell_value	Object	Converted value of a cell, which is used for display purposes.
model_type	String	Attribute type. The options include **measure** (metric), **dimension**, and **metric** (indicator). Enumerated values: • **measure** • **dimension** • **metric**
cell_converted_value	String	Converted value of a cell that is of the String type. It is used to ensure accuracy by converting the value to a String type when the number is too large.

Example Request

Ask data questions in natural language.

```
POST https://{endpoint}/v1/{project_id}/nl-query
{
  "message_content" : "What was the company's revenue in November of last year?",
  "dialog_id" : "66xxxx02x5axxx9xx12fxxx9c2xxxx6d",
  "subject_id" : "xa4axxx4-9xx9-4xx2-9xxx-3axxxxf4a5c3"
}
```

Example Response

Status code: 200

The request for the **nl-query** parameter is successful.

```
{
  "query_data" : [ [ {
```



```
  "caption" : "Revenue",
  "data_type" : "NUMBER",
  "level_type" : null,
  "cell_raw_value" : null,
  "cell_value" : "Revenue",
  "model_type" : "measure",
  "cell_converted_value" : null
 }, {
  "caption" : "Revenue",
  "data_type" : "NUMBER",
  "level_type" : null,
  "cell_raw_value" : "423xxx214.32",
  "cell_value" : "423xxx214.32",
  "model_type" : "measure",
  "cell_converted_value" : null
 } ] ],
 "sql_text" : "SELECT xxx AS xxx FROM xxx.xxx AS xxx",
 "insight" : "Query results: The company's revenue in November of last year was xxx. Suggested areas for exploration: xxx. Business recommendations: xxx."
}
```


Status Codes

Status Code	Description
200	The request for the **nl-query** parameter is successful.

Error Codes

See **Error Codes**.

20 Permissions Policies and Supported Actions

You can use Identity and Access Management (IAM) for fine-grained permissions management of your DataArts Insight. If your Huawei Cloud account does not need individual IAM users, you can skip this section.

New IAM users do not have any permissions assigned by default. You need to first add them to one or more groups and attach policies or roles to these groups. The users then inherit permissions from the groups and can perform specified operations on cloud services based on the permissions they have been assigned.

You can grant users permissions using **roles** and **policies**.

- Roles: Roles are provided by IAM to define service-based permissions that match users' job responsibilities.
- Policies: Policies are more fine-grained, API-based permissions required to perform operations on specific cloud resources under certain conditions, meeting requirements for secure access control.

NOTE

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

An account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions. The required permissions are determined by the actions supported by the API. Only users with the policies allowing for those actions can call the API successfully. For example, if an IAM user needs to create workspaces, the user must have been granted permissions that allow the **dataartsinsight:workspace:create** action.

Supported Actions

DataArts Insight provides system-defined policies that can be directly used in IAM. You can also create custom policies to supplement system-defined policies for more refined access control. Operations supported by policies are specific to APIs. The following are common concepts related to policies:

- Permissions: statements in a policy that allow or deny certain operations.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.

- Actions: specific operations that are allowed or denied.
- Dependencies: actions which a specific action depends on. When allowing an action for a user, you also need to allow any existing action dependencies for that user.
- IAM projects/Enterprise projects: the authorization scope of a custom policy. A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions for both IAM and enterprise projects can be used and applied for both IAM and Enterprise Management. Policies that contain actions only for IAM projects can be used and applied to IAM only.

  For details about the differences between IAM and enterprise projects, see **Differences Between IAM and Enterprise Management**.

NOTE

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

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

Table 20-1 Actions supported by DataArts Insight

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Saving or modifying resource attribute values	POST /v1/{project_id}/authorization/cooperate-authorization/properties	dataartsinsight:resource:auth	-	√	×
Obtaining resource attribute values	GET /v1/{project_id}/authorization/cooperate-authorization/properties	dataartsinsight:resource:auth	-	√	×
Listing authorization rule records	GET /v1/{project_id}/authorization/cooperate-authorization/rules	dataartsinsight:resource:auth	-	√	×

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Saving, modifying, and deleting specified self-developed authorization rule records in batches	POST /v1/{project_id}/authorization/cooperate-authorization/rules/batch-save	dataartsinsight:resource:auth	-	√	×
Creating workspaces	POST /v1/{project_id}/instances/{instance_id}/workspaces	dataartsinsight:workspace:create	-	√	×
Deleting workspaces	DELETE /v1/{project_id}/instances/{instance_id}/workspaces/{workspace_id}	dataartsinsight:workspace:delete	-	√	×
Modifying workspaces	PUT /v1/{project_id}/instances/{instance_id}/workspaces/{workspace_id}	dataartsinsight:workspace:update	-	√	×

21 Public Parameters

21.1 Status Codes

Table 21-1 Return codes for successful requests

Return Code	Type	Description
200	OK	Normal response to GET, PUT, and POST requests
201	Created	Normal response to POST requests
204	No Content	Normal response for DELETE operations

Table 21-2 Return values for failed requests

Return Value	Description
400 Bad Request	The server failed to process the request.
401 Unauthorized	You need to enter the username and password to access the page requested.
403 Forbidden	Access to the requested page is forbidden.
404 Not Found	The server cannot find the requested page.
405 Method Not Allowed	The method specified in the request is not allowed.
406 Not Acceptable	The response generated by the server could not be received by the client.
407 Proxy Authentication Required	You must use a proxy server for authentication so that the request can be processed.

Return Value	Description
408 Request Timeout	The request timed out.
409 Conflict	The request could not be processed due to a conflict.
500 Internal Server Error	The request failed to be fulfilled because of a service error.
501 Not Implemented	The request failed to be fulfilled because the server does not support the required function.
502 Bad Gateway	The request failed to be fulfilled because the server receives an invalid response from an upstream server.
503 Service Unavailable	The request failed to be fulfilled because the system is abnormal.
504 Gateway Timeout	Gateway timed out.

21.2 Error Codes

If an error occurs when calling an API, the HTTP request will return an HTTP status code of 4*xx* or 5*xx*. The response body contains the specific error code and information. If you fail to locate the cause of the error, contact technical support and provide the error code for troubleshooting.

Format of an Error Response Body

If an error occurs during API calling, the system returns an error code and a message to you. The following shows the format of an error response body:

```
{
    "error_msg": "xxx",
    "error_code": "DataArtsInsight.20010003"
}
```

In the preceding information, **error_code** is an error code, and **error_msg** describes the error.

Table 21-3 Exceptions

Parameter	Type	Description
error_code	String	Error code.
error_msg	String	Error details.

Error Code Description

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in **Error Codes**.

Table 21-4 Error codes

Status Code	Error Code	Error Message
-	20010003	There is an issue with IAM permissions.
404	24010003	Resource not found.
400	24010043	The current resource lacks collaborative permissions.
400	24150000	Invalid workspace name. The name can contain a maximum of 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
400	24150001	The workspace name is already taken. Please choose a different name.
400	24150002	The default workspace cannot be deleted.
400	24150003	Operation failed. The maximum number of workspaces has been reached.
400	24150004	Operation failed. Check whether the workspace status is normal.
400	24150005	The workspace does not exist.
400	24150010	The workspace requires members. Please choose members to add.
400	24150011	At least one role must be retained for workspace members.
400	24150012	The creator of the workspace cannot be deleted.
400	24150020	Please select a user group. It cannot be left empty.
400	24150021	The memory storage capacity of the workspace is abnormal.
400	24150022	There are only [{0}] bytes of unused capacity, which is not enough to release [{1}] bytes.

21.3 Obtaining a Project ID

A project ID is required for some URLs when an API is called. Obtain a project ID using either of the following methods:

- **Obtain a Project ID by Calling an API**
- **Obtain a Project ID from the Console**

Obtaining a Project ID by Calling an API

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

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

```
{
    "projects": [
        {
            "domain_id": "xxxx2450e8f64ac0xxx180d14e684b",
            "is_domain": false,
            "parent_id": "xxxx450e8f6xxxx0cd18xxxxx4b",
            "name": "project_name",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4axxxx98fb4474xxxxxd05f897d6b99"
            },
            "id": "axxxx98fb4474fa22cxxxx97d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

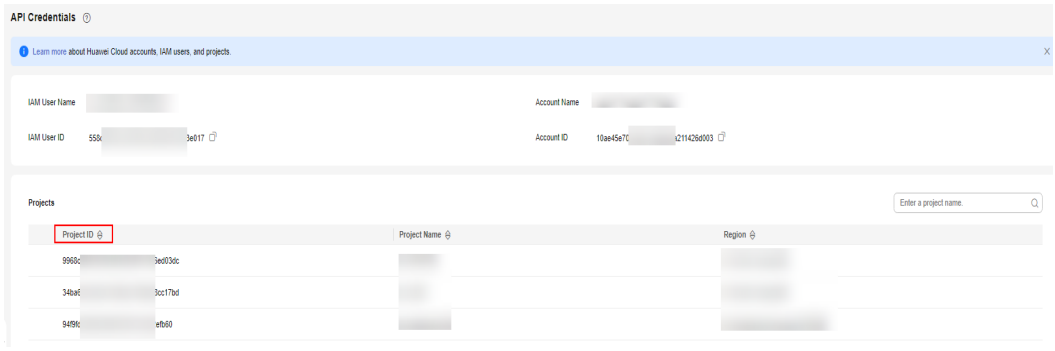
Obtaining a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Log in to the management console.
2. Hover over the username in the upper right corner and select from the drop-down list.

   On the **Projects** tab of the **API Credentials** page, view project IDs.

Figure 21-1 Viewing the project ID



21.4 Obtaining an Account ID

An account ID (domain-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 over the username in the upper right corner and select from the drop-down list.
3. On the page, view **Account ID**.

21.5 Obtaining a Workspace ID

1. Log in to the DataArts Insight console.
2. In the navigation pane on the left, choose **System Settings**. Choose **Organization Management** > **My Projects**.
3. Under **My Projects**, click the project for which you want to view the workspace ID. On the **Basic Information** tab, **ID** is the workspace ID.

   NOTE

   Alternatively, you can query a workspace ID by calling an API. For details, see **Querying Workspaces**.

Figure 21-2 Viewing a workspace ID