Data Ingestion Service

API Reference

Issue 01

Date 2020-07-07

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Contents

1 Before You Start

1.1 Overview

Welcome to *Data Ingestion Service API Reference*. Data Ingestion Service (DIS) provides efficient collection, transmission, and distribution capabilities for real-time IoT and Internet data, supports multiple IoT protocols, and provides various APIs to help you quickly build real-time data applications.

This document describes how to use application programming interfaces (APIs) to perform operations on DIS, such as uploading or downloading data. For details about all supported operations, see **API Overview**.

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

1.2 API Calling

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

1.3 Endpoints

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

1.4 Constraints

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

1.5 Concepts

- Account

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

- IAM User

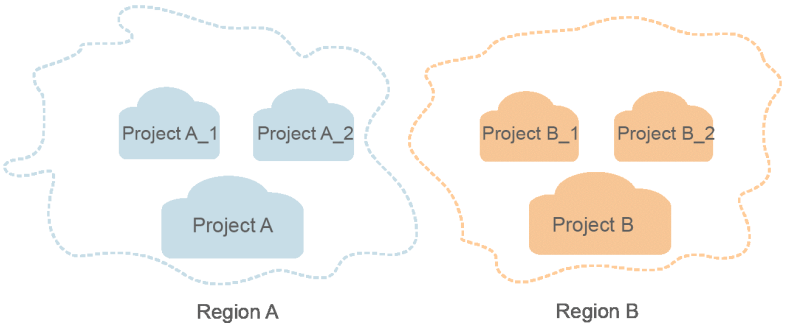
  An IAM User is created using an accounta domain to use cloud services. Each IAM user has its own identity credentials (password and access keys).

  An IAM user can view the account ID and user ID on the **My Credentials**.The account, username, and password will be required for API authentication.

- Project

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

  Figure 1-1 Project isolating model

- Checkpoint

  When an application consumes data, the latest SN of the consumed data is recorded as a checkpoint. When the data is reconsumed, the consumption can be continued based on this checkpoint.

- Application

  Multiple applications can consume data in the same stream. The consumed data in the stream by each application is recorded by checkpoints generated for each application.

2 API Overview

The DIS API is a self-developed API that complies with REST API design specifications. DIS provides the functions listed in **Table 2-1** through the DIS API.

Table 2-1 API functions

API Name	Function	API URI
Stream management	**Creating a DIS Stream**	POST /v2/{project_id}/streams
	Deleting a DIS Stream	DELETE /v2/{project_id}/streams
	Listing DIS Streams	GET /v2/{project_id}/streams
	Viewing Details of a DIS Stream	GET /v2/{project_id}/streams/{stream_name}
	Querying Stream Monitoring Information	GET /v2/{project_id}/streams/{stream_name}/metrics?label={label}&start_time={start_time}&end_time={end_time}
	Obtaining Stream Consumption Information	GET /v2/{project_id}/apps/{app}/streams/{stream_name}
Data management	**Uploading Data to a DIS Stream**	POST /v2/{project_id}/records
	Downloading Data from a DIS Stream	GET /v2/{project_id}/records{?partition-cursor}
	Obtaining a Cursor	GET /v2/{project_id}/cursors{?stream-name,partition-id,cursor-type,starting-sequence-number}
Program management	**Creating an Application**	POST /v2/{project_id}/apps

API Name	Function	API URI
	Deleting an Application	DELETE /v2/{project_id}/apps/{app_name}
	Querying an Application List	GET /v2/{project_id}/apps
	Viewing Details About a Consumer Application	GET /v2/{project_id}/apps/{app_name}
Checkpoint management	**Adding a Checkpoint**	POST /v2/{project_id}/checkpoints
	Querying a Checkpoint	GET /v2/{project_id}/checkpoints{?stream_name,partition_id,app_name,checkpoint_type}
	Deleting a Checkpoint	DELETE /v2/{project_id}/checkpoints
Dump task management	**Adding a Dump Task**	POST /v2/{project_id}/streams/{stream_name}/transfer-tasks
	Deleting a Dump Task	DELETE /v2/{project_id}/streams/{stream_name}/transfer-tasks/{transfer_task_name}
	Querying Dump Task Details	GET /v2/{project_id}/streams/{stream_name}/transfer-tasks/{transfer_task_name}
	Querying a Dump Task List	GET /v2/{project_id}/streams/{stream_name}/transfer-tasks

3 Calling APIs

3.1 Making an API Request

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

Request URI

A request URI is in the following format:

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

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

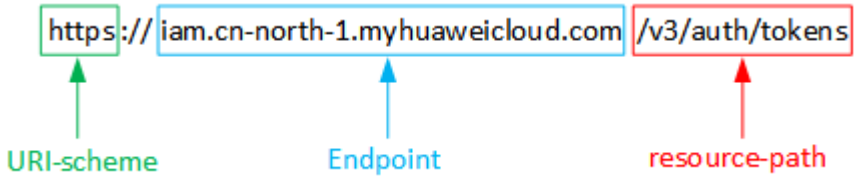
Table 3-1 URI parameter description

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of IAM in the **CN North-Beijing4** region is **iam.cn-north-4.myhuaweicloud.com**.
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.
qcnuery-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 North-Beijing4** region, obtain the endpoint of IAM **(iam.cn-north-4.myhuaweicloud.com)** for this region and the **resource-path (/v3/auth/tokens)** in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-1 Example URI



NOTE

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

Request Methods

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

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

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

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


Request Header

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

Table 3-3 lists common request headers.

Table 3-3 Common request headers

Parameter	Description	Mandatory	Example
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of *Hostname:Port number*. If the port number is not specified, the default port is used. The default port number for HTTPS is 443.	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the request body MIME type. Its default value is **application/json**. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No	e9993fc787d94b6c886cbaa340f9c0f4

Parameter	Description	Mandatory	Example
X-Auth-Token	Specifies the user token. It is a response to the API used to **obtain a user token**. This API is the only one that does not require authentication. After the request is processed, the value of **X-Subject-Token** in the header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQc-Co...ggg1BBIINPXsidG9rZ

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.

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

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

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

(Optional) Request Body

This part is optional. The body of a request is often sent in a structured format as specified in the **Content-Type** header field, such as JSON or XML. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to **obtain a user token**, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*, *domainname*, ******** (login password), and *xxxxxxxxxxxxxxxxx* (project ID) with the actual values. To learn how to obtain a project ID, see **Obtaining a Project ID**.

NOTE

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

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxxxxxxxxxxx"
            }
        }
    }
}
```

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

3.2 Authentication

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

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

Token-based Authentication

NOTE

- The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.
- DIS is a project-level service deployed in specific physical regions. When obtaining a token, set **scope** in the message body to **project**, indicating that the obtained token can be used to access only resources in a specific project. **project** supports parameters **ID** and **name**. You can set either of them.

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

When calling an API to **Obtain a User Token**, you must set **auth.scope** in the request body to **project**.

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

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

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

AK/SK-based Authentication

NOTE

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

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

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

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

NOTICE

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

3.3 Response

Status Code

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

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

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

Response Header

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

Figure 3-2 shows the response header fields for the API used to **obtain a user token**. The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

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

(Optional) Response Body

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

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

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

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

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

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

3.4 AK/SK Signature Authentication Methods

3.4.1 Authentication of Signature in a Header

For all API operations, the most common identity authentication is to carry signatures in headers.

In the header, the signature is carried in the Authorization header field of the HTTP message. The format of the message header is as follows:

Authorization: SDK-HMAC-SHA256 Credential=ak/CredentialScope, SignedHeaders=SignedHeaders, Signature=signature

Parameters to be Prepared before Signing a Signature

Table 3-4 Parameters related to the request signature

Parameter	Description	Example
method	Request type.	POST
ak	User AK.	DJZN5UEQSODCWJ7NGOMC
sk	User SK.	vRNwGMd92PlityIO3daDseoS9hciL9xKSKkBiJ44
projectId	User project ID.	d575b0b740e54221aeb9a165653b103d

Parameter	Description	Example
region	Information about the region where the service is located (cn-north-4 indicates CN North-Beijing4)	-
url	Request URL.	https://dis.${region}.myhuaweicloud.com:20004/v2/d575b0b740e54221aeb9a165653b103d/records/?stream-name=test2&partition-id=0
content	Request body.	{"stream_name":"test2","records":[{"data":"aGVsbG8gd29ybGQu","partition_id":"","explicit_hash_key":"","partition_key":"0"}]}
service	Service abbreviation.	dis
header	HTTP request header (The request header in the signature is not mandatory. You can add a user-defined request header after the signature is complete. If there is a request header before the signature, all request headers need to be put into the signature calculation process during the signature.)	-

3.4.2 Signature Process

This topic describes how to generate a signature.

- Step 1: **Adding New Request Headers to the Request Header List**
- Step 2: **Calculating the SHA-256 Value of the Request Body**
- Step 3: **Calculating CanonicalRequest**
- Step 4: **Calculating StringToSign**
- Step 5: **Calculating SigningKey**
- Step 6: **Calculating Signature**
- Step7: **Generating Signature Information for Requests**

NOTE

The signature result varies with time. In the following example, the value of **X-Sdk-Date** is **20181101T081630Z**. If the signature result of the code is the same as the sample result, do not obtain the value of **X-Sdk-Date** based on the current time, the value is fixed to **20181101T081630Z**. In this way, the signature results can be compared. After the test is complete, change the value of **X-Sdk-Date** based on the current time.

Adding New Request Headers to the Request Header List

Table 3-5 Descriptions of the request header

Request Header	Description	Example
Host	Domain name and port number in the URL. The format is *host:port* (that is, domain name: port number). **NOTE** If any of the following conditions is met, delete *:port* and reserve the host: • The URL does not contain any specified port (the default port is used). • The HTTP protocol is used and port 80 is specified. • If the HTTPS protocol is used and port 443 is specified.	• If the URL is **https://dis.${region}.myhuaweicloud.com/v2/...**, the value of **Host** is **dis.${region}.myhuaweicloud.com**. • If the URL is **https://dis.${region}.myhuaweicloud.com:443/v2/...**, the value of **Host** is **dis.${region}.myhuaweicloud.com**. • If the URL is **https://dis.${region}.myhuaweicloud.com:20004/v2/...**, the value of **Host** is **dis.${region}.myhuaweicloud.com:20004**. • If the URL is **https://114.115.204.40/v2/...**, the value of **Host** is **114.115.204.40**.

Request Header	Description	Example
X-Sdk-Date	Character string generated based on the current UTC time (Beijing time - 8 hours). The format is yyyyMMdd**T**HHmmss**Z**. YYYY is the year, MM is the month, dd is the day, T is the fixed character, HH is the 24-hour clock, mm is the minute, ss is the second, and Z is the fixed character. For example, on 2018-11-01 16:16:30 (Beijing time), the generated **X-Sdk-Date** value is 20181101T081630Z.	20181101T081630Z

The request header list generated based on the example is as follows:

```
Host: dis.${region}.myhuaweicloud.com
X-Sdk-Date: 20181101T081630Z
```

Calculating the SHA-256 Value of the Request Body

String calculateContentHash=sha256(content)

The value is obtained by calculating the SHA-256 value of the request body content.

NOTICE

If there is no request body (that is, the value of content is null), the value of calculateContentHash is fixed to **e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855**.

The following is an example:

{"stream_name":"test2","records":
[{"data":"aGVsbG8gd29ybGQu","partition_id":"","explicit_hash_key":"","partition_key":"0"}]},

The calculated calculateContentHash result is as follows:

af22378806bf4e69f5f1667877906e6ead78080cd859b4988ea6714dba6d1e02

Calculating CanonicalRequest

```
String CanonicalRequest =
        HTTPRequestMethod + '\n' +
        CanonicalURI + '\n' +
        CanonicalQueryString + '\n' +
        CanonicalHeaders + '\n' +
        SignedHeaders + '\n' +
        calculateContentHash
```

Table 3-6 Parameters required for constructing CanonicalRequest

Parameter	Description	Example
HTTPRequestMethod	HTTP request type.	POST
CanonicalURI	Resource path between a URL domain name port and a parameter. **NOTICE** If the resource path does not end with a slash (/), add / at the end.	• If the URL is **https://dis.${region}.myhuaweicloud.com/v2/d575b0b740e54221aeb9a165653b103d/records**, the value is **/v2/d575b0b740e54221aeb9a165653b103d/records/**. • If the URL is **https://dis.${region}.myhuaweicloud.com/v2/d575b0b740e54221aeb9a165653b103d/cursors/?stream-name=test2**, the value is **/v2/d575b0b740e54221aeb9a165653b103d/cursors/**.

Parameter	Description	Example
CanonicalQueryString	Character string generated after the query parameters in the URL are sorted in lexicographical order. The format is as follows: CanonicalQueryEntry0 + "&" + CanonicalQueryEntry1 + ... If there is no parameter, the value is an empty character string. **NOTE** The format of CanonicalQueryEntry is UrlEncode(QueryName) + '=' + UrlEncode(QueryValue). That is, URL encoding is performed on the request header and value, and the equal sign (=) is used to connect the request header and value.	partition-id=0&stream-name=test2
CanonicalHeaders	Character string generated after the request headers (case insensitive) are sorted in lexicographical order. The format is as follows: CanonicalHeadersEntry0 + CanonicalHeadersEntry1 + ... **NOTE** The format of CanonicalHeadersEntry is Lowercase(HeaderName) + ':' + Trimall(HeaderValue) + '\n'	host:dis.${region}.myhuaweicloud.com:20004\nx-sdk-date:20181101T081630Z\n

Parameter	Description	Example
SignedHeaders	Character string generated after all the request headers are converted to lowercase letters and are sorted in lexicographical order. The format is Lowercase(HeaderName 0) + ';' + Lowercase(HeaderName 1) + ";" + ... **NOTE** Convert all request header names to lowercase letters, sort them in lexicographical order, and use semicolons (;) to separate multiple lowercase request headers.	host;x-sdk-date
calculateContentHash	SHA-256 value of the request body. For details, see **Calculating the SHA-256 Value of the Request Body**.	af22378806bf4e69f5f1667877906e6ead78080cd859b4988ea6714dba6d1e02

The CanonicalRequest result calculated based on the example is as follows: (\n is displayed as a newline character.)

```
POST
/v2/d575b0b740e54221aeb9a165653b103d/records/
partition-id=0&stream-name=test2
host:dis.${region}.myhuaweicloud.com
x-sdk-date:20181101T081630Z
host;x-sdk-date
af22378806bf4e69f5f1667877906e6ead78080cd859b4988ea6714dba6d1e02
```

If \n is not displayed as a newline, the format is as follows:

```
POST\n/v2/d575b0b740e54221aeb9a165653b103d/records/\npartition-id=0&stream-name=test2\nhost:dis.${region}.myhuaweicloud.com\nx-sdk-date:20181101T081630Z\n\nhost;x-sdk-date
\naf22378806bf4e69f5f1667877906e6ead78080cd859b4988ea6714dba6d1e02
```

Calculating StringToSign

```
String StringToSign  =
    Algorithm + '\n' +
    RequestDate + '\n' +
    CredentialScope + '\n' +
    HashedCanonicalRequest))
```


Table 3-7 Parameters required for constructing StringToSing

Parameter	Description	Example
Algorithm	Encryption algorithm. The value is fixed to **SDK-HMAC-SHA256**.	SDK-HMAC-SHA256
RequestDate	Value of **X-Sdk-Date**.	20181101T081630Z
CredentialScope	Authentication scope. The format is date/region/service/terminationString. • date: the date field in **X-Sdk-Date**. The format is **yyyyMMdd**. • region: the region where the service is provided. • Service: service name. • TerminationString: terminal identifier. The value is fixed to **sdk_request**.	20181101/${region}/dis/sdk_request
HashedCanonicalRequest	SHA-256 value of CanonicalRequest calculated in **Calculating CanonicalRequest**.	bf0eb8735b561a700b85b1142eb61df06569dffcd1088a7dda539e2ee6497809

The following is a sample result:

StringToSign=SDK-HMAC-SHA256\n20181101T081630Z\n20181101/${region}/dis/sdk_request
\n548470a57f61f5841c6869cd51164be0da033c14a874ff7a498593a4ae202b41

Calculating SigningKey

SigningKey is a byte array calculated using the HmacSHA256 algorithm. The method is as follows:

```
kSecret = Bytes("SDK"+sk)
kDate = HmacSHA256(kSecret, Date)
kRegion = HmacSHA256(kDate, Region)
kService = HmacSHA256(kRegion, Service)
SigningKey = HmacSHA256(kService, "sdk_request")
```

The HmacSHA256 algorithm uses a key (in the format of byte array) and a message (in the format of character string) as input to generate a message digest (in the format of byte array) as the output. The result of HmacSHA256 is a byte array. To intuitively show the result of each step and facilitate verification, the following example shows that the result of each step is a hexadecimal character

string, that is, HexEncode(Result), note that the calculation process is still a byte array rather than the value after HexEncode.

Table 3-8 Parameters required for constructing StringKey

Parameter	Description	Example (Actual Value After HexEncode)
kSecret	"SDK"+sk, which is converted into a byte array as the initial key.	53444b76524e77474d643932506c697479494f33646144736 56f5339686369 4c39784b534b6b42694a3434
kDate	HmacSHA256 value of **kSecret** and **Date** (the Date field of **X-Sdk-Date**).	305758792674e5cfec8609daf3725e37367d8479ee824d2914db63004b5211b2
kRegion	HmacSHA256 value of **kDate** and **Region**.	c56298c0270a63bb57779cdfe02d41b55393f8b61bf4c793b06866c14f9b28e7
kService	HmacSHA256 value of **kRegion** and **Service**.	ed5246fb17c384c46000ba85a7c788e3e18c5e0323240f9bff6a1308df9179e8
SigningKey	HmacSHA256 value of **kService** and **sdk_request**.	1ea4929f7f18601abb9af0aaa9dc46eb0b6bda7b1de20d2a152dbe76e05dffad

The StringToSign result is a byte array and cannot be directly represented in the document. Therefore, StringToSign is represented by a hexadecimal string (that is, HexEncode(StringToSign), which is the following character string:

```
1ea4929f7f18601abb9af0aaa9dc46eb0b6bda7b1de20d2a152dbe76e05dffad
```

NOTICE

The character string is not the result of StringToSign but the result of HexEncode(StringToSign) for the convenience of the user to check whether the result is correct. When Signature is calculated in the next step, the value of StringToSign must be transferred.

NOTE

SigningKey is related only to **sk**, **date**, **region**, and **service**. When the four parameters remain unchanged, the value of SigningKey is fixed. For better performance, the SigningKey result can be cached and used repeatedly.

Calculating Signature

```
String Signature = HexEncode(HmacSHA256(SigningKey, StringToSign))
```

Parameters required for constructing Signature

The value of **String Signature** is a hexadecimal character string converted from the HmacSHA256 value of **SigningKey** and **StringToSign**.

The Signature result calculated based on the example is as follows:

```
8df520f285a18b7b101fc0d6507de03c4078460c65baa289ffa49ca718e9190b
```

Generating Signature Information for Requests

After the following value is added to a request header, the signing is completed:

Authorization: SDK-HMAC-SHA256 Credential=ak/CredentialScope, SignedHeaders=SignedHeaders, Signature=signature

- ak: user AK.

  Sample: DJZN5UEQSODCWJ7NGOMC
- CredentialScope: the authentication range used in **Calculating StringToSign**.

  Example: 20181101/${region}/dis/sdk_request
- SignedHeaders: SignedHeaders used in **Calculating CanonicalRequest**.

  Sample: host;x-sdk-date
- Signature: the signature obtained in **Calculating Signature**.

  Example: 8df520f285a18b7b101fc0d6507de03c4078460c65baa289ffa49ca718e9190b

  The signature request header calculated based on the example is as follows:

  ```
  Authorization: SDK-HMAC-SHA256 Credential=DJZN5UEQSODCWJ7NGOMC/20181101/${region}/dis/sdk_request, SignedHeaders=host;x-sdk-date, Signature=8df520f285a18b7b101fc0d6507de03c4078460c65baa289ffa49ca718e9190b
  ```

Problem Analysis

- If the signature is correct but the AK or project ID does not exist, the server returns the following information:

  ```
  441 : Invalid AccessKey header. [Invaild ak.]
  ```

  Handling method:

  Ensure that the AK is correct by referring to **AK/SK-based Authentication**.

  Ensure that the Project ID is correct by referring to **Obtaining a Project ID**.
- If the signature logic or SK is incorrect, the server returns the following information:

  ```
  441 : Invalid authorization request.
  ```

  Handling method:

  Use the sample parameters to verify the signature process and compare the verification result with the sample result to ensure that the signature process is correct.
- If the region is incorrect (for example, send a request for a DIS address in CN South ${region} when the current region is CN North1 ${region}, the server returns the following information:

  ```
  441 : Invalid Region header. [${region}]
  ```


Handling method:

Ensure that the region maps the DIS address by referring to **Obtaining a Project ID**.

- If the value of **X-Sdk-Date** is more than 15 minutes later than the standard time, the server returns the following information:

```
441 : Invalid X-Sdk-Date header
```

Handling method:

Adjust the client time to the standard time of the corresponding time zone.

4 Getting Started

This section describes how to create a DIS stream by calling APIs.

NOTE

The token obtained on IAM is valid for only 24 hours. If you want to use one token for authentication, you can cache it to avoid frequent calling.

Involved APIs

If you use a token for authentication, you must obtain the token and add **X-Auth-Token** to the request header of the API when making an API call.

- API for obtaining tokens from IAM
- API for creating a DIS stream

Procedure

1. Obtain the token by following instructions in **Token-based Authentication**.
2. Send **POST https://Endpoint of DIS/v2/{project_id}/streams**.
3. Add **X-Auth-Token** to the request header.
4. Specify the following parameters in the request body:

```
{
    "stream_name": "dis-DLpR",
    "partition_count": 1,
    "stream_type": "COMMON",
    "data_type": "BLOB",
    "data_duration": 24
}
```

If the request is successful, 201 Created is returned.

If the request fails, an error code and error information are returned. For details, see section **Error Codes**.

5 API Description

5.1 Creating a DIS Stream

Function

This API is used to create a stream.

- When creating a channel, you need to specify the stream type (common or advanced), number of partitions, and source data type.
- A maximum of 10 advanced stream partitions and 50 common stream partitions can be created for an account by default. You can **submit a service ticket** to increase the quota.

URI

- URI format

  POST /v2/{project_id}/streams
- Parameter description

  None

Request

- Example request
  - Create a stream whose source data type is BLOB.

    ```
    POST https://{endpoint}/v2/{project_id}/streams
    {
       "stream_name": "dis-DLpR",
       "partition_count": 1,
       "stream_type": "COMMON",
       "data_type": "BLOB",
       "data_duration": 24
    }
    ```
  - Create a stream whose source data type is JSON or CSV.

    ```
    POST https://{endpoint}/v2/{project_id}/streams
    {
       "stream_name": "dis-DLpR",
       "partition_count": 1,
       "stream_type": "COMMON",
    ```



```
    "data_type": "JSON",
  "data_duration": 24,
    "data_schema":"{\"type\":\"record\",\"name\":\"RecordName\",\"fields\":[{\"name\":\"id\",
\"type\":\"string\",\"doc\":\"Type inferred from '\\\"2017/10/11 11:11:11\\\"'\"},{\"name\":\"info
\",\"type\":{\"type\":\"array\",\"items\":{\"type\":\"record\",\"name\":\"info\",\"fields\":[{\"name
\":\"date\",\"type\":\"string\",\"doc\":\"Type inferred from '\\\"2018/10/11 11:11:11\\\"'\"}]}},
\"doc\":\"Type inferred from '[{\\\"date\\\":\\\"2018/10/11 11:11:11\\\"}]'\"}]}"}
}
```

- Parameter description

Table 5-1 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream. Each DIS stream has a unique name. A stream name is 1 to 64 characters in length. Only letters, digits, hyphens (-), and underscores (_) are allowed.
stream_type	No	String	Stream type. Possible values: ● **Common**: a common stream. The bandwidth is 1 MB/s. ● **ADVANCED**: an advanced stream. The bandwidth is 5 MB/s. Default value: COMMON
partition_count	Yes	Int	Quantity of the partitions into which data records in the newly created DIS stream will be distributed. Partitions are the base throughput unit of a DIS stream. The value range varies depending on the value of **stream_type**. ● If **stream_type** is not specified or set to **COMMON**, the value of **partition_count** is an integer from 1 to 50. If the tenant has created **N** common partitions, the maximum value of **partition_count** is **50-N**. ● If **stream_type** is set to **ADVANCED**, the value of **partition_count** is an integer from 1 to 10. If the tenant has created **N** advanced partitions, the maximum value of **partition_count** is **10-N**.

Parameter	Mandatory	Type	Description
data_duration	No	Int	Period of time for which data is retained in the DIS stream. Value range: 24 to 72 Unit: hour Default value: 24 If this parameter is left unspecified, the default value will be used.
data_type	No	String	Source data type. Possible values: • **BLOB**: a collection of binary data stored as a single entity in a database management system. • **JSON**: an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types. • **CSV**: a simple text format for storing tabular data in a plain text file. Default value: **BLOB**
data_schema	No	String	Source data structure that defines JOSN and CSV formats. It is described in the syntax of Avro. For details about Avro, see **http://avro.apache.org/docs/current/#schemas**. **NOTE** This parameter is mandatory when **Dump File Format** is **Parquet** or **CarbonData**.

Response

- If the DIS stream was successfully created, a 201 response with an empty response body is returned.
- If the DIS stream failed to be created, identify the failure cause according to the response body and the instructions in **Error Codes**.

Status Code

- Normal

  201 Created
- Failed

  For more information, see **Error Codes**.

5.2 Deleting a DIS Stream

Function

This API is used to delete a DIS stream after pushing or pulling data to or from the stream or if the stream does not run properly.

Before deleting a stream, you need to specify the stream name.

URI

- URI format

  DELETE /v2/{project_id}/streams/{stream_name}
- Parameter description

  None

Request

- Example request
  ```
  DELETE https://{endpoint}/v2/{project_id}/streams/stream_test
  ```
- Parameter description

  None

Response

- If the DIS stream was successfully deleted, a 204 response with an empty response body is returned. For details about status codes, see **Status Codes**.
- If the DIS stream failed to be created, identify the failure cause according to the response body and the instructions in **Error Codes**.

Status Code

- Normal

  204 NOCONTENT
- Failed

  For more information, see **Error Codes**.

5.3 Listing DIS Streams

Function

This API is used to query all streams created by the current tenant.

During query, you need to specify the stream from which the stream list is returned and the maximum number of streams returned in a single request.

URI

- URI format

  GET /v2/{project_id}/streams

- Parameter description

  None

Request

- Example request

```
GET https://{endpoint}/v2/{project_id}/streams?limit=10&start_stream_name=test_stream
```

- Parameter description

Table 5-2 Parameter description

Parameter	Mandatory	Type	Description
limit	No	Int	The maximum number of DIS streams to list in a single API call. Value range: 1 to 100 Default value: 10
start_stream_name	No	String	Name of the DIS stream to start the list with. The returned stream list does not contain this DIS stream name. If pagination query is required, this parameter is not transferred when you query data on the first page. If the value of **has_more_streams** is **true**, go to the next page, **start_stream_name** is set to the name of the last stream in the query result on the first page. **NOTE** Streams in the stream list are arranged in alphabetical order.

Response

- Example response

```
{
  "total_number": 1,
  "has_more_streams": false,
  "stream_names": [
    "stream_test"
  ]
}
```

- Parameter description

Table 5-3 Response parameter description

Parameter	Type	Description
total_number	Int	Total number of DIS streams created by the current tenant.
stream_names	List<String>	Name of the list of the streams meeting the current requests.
has_more_streams	Boolean	Specify whether there are more matching DIS streams to list. Possible values: • **true**: There are more streams. • **false**: There are no more streams.

Status Code

- Normal
  200 OK
- Failed
  For more information, see **Error Codes**.

5.4 Viewing Details of a DIS Stream

Function

This API is used to query details about a specified stream.

- You need to specify the name of the stream to be queried.
- You need to specify the partition from which the partition list is returned and the maximum number of partitions returned in a single request.

URI

- URI format
  GET /v2/{project_id}/streams/{stream_name}
- Parameter description
  None

Request

- Example request
  ```
  GET https://{endpoint}/v2/{project_id}/streams/stream_test?
  start_partitionId=shardId-0000000000&limit_partitions=100
  ```
- Parameter description

Table 5-4 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream to be queried.
start_partitionId	No	String	ID of the partition to start the partition list with. The returned partition list does not contain this partition ID.
limit_partitions	No	Int	Maximum number of partitions to list in a single API call. Value range: 1–1000 Default value: 100

Response

- Example response

```
{
  "stream_name": "stream_test",
  "stream_id": "U7v0U582F8ccXyErJKC",
  "create_time": 1504679587519,
  "last_modified_time": 1504679587519,
  "retention_period": 24,
  "status": "RUNNING",
  "stream_type": "COMMON",
  "data_type": "JSON",
  "writable_partition_count": 3,
  "readable_partition_count": 5,
  "partitions": [
    {
     "status": "ACTIVE",
     "partition_id": "shardId-0000000000",
      "hash_range": "[0 : 9223372036854775807]",
       "sequence_number_range": "[0 : 200]"
    }
 ],
 "data_schema":{
    "type": "record",
    "name": "RecordName",
    "fields": [{
        "name": "id",
        "type": "string",
        "doc": "Type inferred from '\"1\"'"
              }, {
        "name": "detail",
        "type": {
            "type": "record",
            "name": "detail",
            "fields": [{
                "name": "detID",
                "type": "string",
                "doc": "Type inferred from '\"05790110000000000103\"'"
            }, {
                "name": "endTime",
                "type": "string",
                "doc": "Type inferred from '\"2018/10/07 13:26:35\"'"
            }]
        },
```



```
      "doc": "Type inferred from '{\"detID\":\"05790110000000000103\",\"endTime\":\"2018/10/07
13:26:35\"}'"
    }]
  },
  "has_more_partitions": false,
  "tags": []
}
```

- Parameter description

Table 5-5 Response parameter description

Parameter	Type	Description
stream_name	String	Name of the DIS stream.
stream_id	String	Unique ID of each stream.
create_time	Long	Timestamp at which the DIS stream was created.
last_modified_time	Long	Timestamp at which the DIS stream was most recently modified.
retention_period	Int	Period for storing data in units of hours.
status	String	The stream status is one of the following: • CREATING • RUNNING • TERMINATING • TERMINATED
stream_type	String	Partition type.
data_type	String	Type of the source data.
data_schema	String	Source data structure that defines JSON and CSV formats. It is described in the syntax of the Avro schema. For details about Avro, see **http://avro.apache.org/docs/current/#schemas**.
writable_partition_count	Int	Total number of writable partitions (including partitions in ACTIVE state only).
readable_partition_count	Int	Total number of readable partitions (including partitions in ACTIVE and DELETED state).
tags	List<Tag>	Label of the stream.
partitions	List<PartitionResult>	A list of partitions that comprise the DIS stream. For more information, see **Table 5-6**.

Parameter	Type	Description
has_more_partitions	Boolean	Specify whether there are more matching partitions of the DIS stream to list. • **true**: There are more partitions. • **false**: There are no more partitions.

Table 5-6 partitions parameter description

Parameter	Type	Description
status	String	Current status of each partition. • CREATING • ACTIVE • DELETED • EXPIRED
partition_id	String	Unique identifier of the partition.
hash_range	String	Possible value range of the hash key used by each partition.
sequence_number_range	String	Sequence number range of each partition.

Table 5-7 tags parameter description

Parameter	Mandatory	Type	Description
key	Yes	String	Key. A tag key cannot contain special characters such as =*<>\,|/ or start or end with a space.
value	Yes	String	Value. A tag value cannot contain special characters such as =*<>\,|/ or start or end with a space.

Status Code

- Normal
  200 OK
- Failed
  For more information, see **Error Codes**.

5.5 Uploading Data to a DIS Stream

Function

This API is used to upload data to a DIS stream.

- Before uploading data, you need to create a stream by referring to **Creating a DIS Stream**.
- When uploading data, you need to specify the partition ID of the stream. The partition ID can be obtained from **Viewing Details of a DIS Stream**.

URI

- URI format

  POST /v2/{project_id}/records
- Parameter description

  None

Request

- Example request

```
POST https://{endpoint}/v2/{project_id}/records

{
 "stream_name": "stream_test",
  "records": [
    {
      "data":
"MTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTE=",
      "explicit_hash_key": null,
      "partition_key": "0"
 ]
}
```

- Parameter description

Table 5-8 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the stream created on the management console. A stream name is 1 to 64 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Mandatory	Type	Description
stream_id	No	String	Unique ID of the stream. If no stream is found based on stream_name and stream_id is not empty, stream_id is used to search for the stream. **NOTE** This parameter is mandatory when you upload data to a stream that has been authorized.
records	Yes	List<PutRecordsRequestEntry>	Information of data records. For more information, see **Table 5-9**.

Table 5-9 records parameter description

Parameter	Mandatory	Type	Description
data	Yes	Base64-encoded binary data object	Data to be put into the chosen DIS stream. The uploaded data must be serialized into binary Base64-encoded data.
explicit_hash_key	No	String	Hash value used to explicitly determine the partition into which an individual data record will be put. The hash value overrides the partition key hash. Value range: 0 to long.max
partition_id	No	String	Unique identifier of the partition.
partition_key	No	String	Partition key of the partition into which an individual data record will be put. **NOTE** The **partition_id** parameter takes precedence over the **partition_key** parameter. If the **partition_id** parameter is not passed in, then the **partition_key** parameter is selected for use.

NOTE

- The user's raw data may contain invisible characters. Therefore, the raw data must be encoded using Base64 encoding before invoking the API.
- If a user uploads data using SDK provided by DIS, the SDK will automatically encode the raw data using Base64 encoding.

Response

- Example response

```
{
  "failed_record_count": 0,
  "records": [
    {


      "sequence_number": "195",
      "partition_id": "shardId-0000000000"
    }
  ]
}
```

- Parameter description

Table 5-10 Response parameter description

Parameter	Type	Description
failed_record_count	Int	The number of data records that failed to be put into the selected DIS stream.
records	List<PutRecordResult>	Processing result of each data record. For more information, see **Table 5-11**.

Table 5-11 PutRecordResult parameter description

Parameter	Type	Description
error_code	String	Error code for an individual record result.
error_message	String	Error message for an individual record result.
partition_id	String	Partition ID for an individual record result.
sequence_number	String	Sequence number of an individual data record. Each data record has a sequence number that is unique within its partition. The sequence number is assigned by DIS when a data producer calls PutRecords to add data to a DIS stream. Sequence numbers for the same partition key generally increase over time; the longer the time period between write requests (PutRecords requests), the larger the sequence numbers become.

Status Code

- Normal

  200 OK
- Failed

  For more information, see **Error Codes**.

5.6 Obtaining a Cursor

Function

This API is used to obtain a data cursor.

There are five cursor types available:

- AT_SEQUENCE_NUMBER
- AFTER_SEQUENCE_NUMBER
- TRIM_HORIZON
- LATEST
- AT_TIMESTAMP

To download streaming data, you need to determine the position where the data is obtained from the partition, that is, to obtain the cursor. After the start position is determined, the data is obtained cyclically.

URI

- URI format

  GET /v2/{project_id}/cursors{?stream-name,stream-id,partition-id,cursor-type,starting-sequence-number}
- Parameter description

  None

Request

- Example request

  ```
  GET https://{endpoint}/v2/{project_id}/cursors?stream-name=lzc08&stream-id=i9eOJp4sTYRApsWwJDS&partition-id=0&cursor-type=AT_SEQUENCE_NUMBER&starting-sequence-number=l1
  ```
- Parameter description

Table 5-12 Parameter description

Parameter	Mandatory	Type	Description
stream-name	Yes	String	Name of the stream created on the management console. Value range: Only letters, digits, hyphens (-), and underscores (_) are allowed. The value must be 1 to 64 characters long.
stream_id	No	String	Unique ID of the stream. If no stream is found based on stream_name and stream_id is not empty, stream_id is used to search for the stream.
partition-id	Yes	String	Partition ID of a stream. Two partition ID formats are available: • shardId-0000000000 • 0

Parameter	Mandatory	Type	Description
cursor-type	No	String	Cursor type. • AT_SEQUENCE_NUMBER: Data is read from the position denoted by a specific sequence number (that is defined by **starting-sequence-number**). This is the default cursor type. • AFTER_SEQUENCE_NUMBER: Data is read right after the position denoted by a specific sequence number (that is defined by **starting-sequence-number**). • TRIM_HORIZON: Data is read from the earliest valid records that are stored in the partition. For example, a tenant uses a DIS stream to upload three pieces of data A1, A2, and A3. N days later, A1 has expired and A2 and A3 are still in the validity period. In this case, if the tenant uses TRIM_HORIZON to download the data, only A2 and A3 are downloadable. • LATEST: Data is read just after the most recent record in the partition. This setting ensures that you always read the most recent data in the partition. • AT_TIMESTAMP: Data is read from the position denoted by a specific timestamp (that is defined by **timestamp**).

Parameter	Mandatory	Type	Description
starting-sequence-number	No	String	Sequence number of an individual data record. Each data record has a sequence number that is unique within its partition. The sequence number is assigned by DIS when a data producer calls PutRecords to add data to a DIS stream. Sequence numbers for the same partition key generally increase over time; the longer the time period between write requests (PutRecords requests), the larger the sequence numbers become. The sequence number is closely related to cursor types **AT_SEQUENCE_NUMBER** and **AFTER_SEQUENCE_NUMBER**. The two parameters determine the location of the data to be accessed. Value range: 0 to 9223372036854775807
timestamp	No	Long	Timestamp when the data record starts to be accessed. It is closely related to cursor type **AT_TIMESTAMP**. The two parameters determine the location of the data to be accessed. **NOTE** This timestamp is accurate to milliseconds.

Response

- Example response

```
{
 "partition_cursor":
"eyJnZXRJdGVyYXRvclBhcmFtIjp7InN0cmVhbS1uYW1lIjoianpjIiwicGFydGl0aW9uLWlkIjoiMCIsImN1cnNvci10eXBlIjoiQVRfU0VRVUVOQ0VfTlVNQkVSIiwic3RhcnRpbmctc2VxdWVuY2UtbnVtYmVyIjoiMTAifSwiZ2VuZXJhdGVUaW1lc3RhbXAiOjE1MDYxNTk1NjM0MDV9"
}
```

- Parameter description

Table 5-13 Response parameter description

Parameter	Type	Description
partition_cursor	String	Cursor of the partition used to specify the position in the partition from which to start reading data records sequentially. Value: 1 to 512 characters **NOTE** The validity period of a cursor is 5 minutes.

Status Code

- Normal

  200 OK
- Failed

  For more information, see **Error Codes**.

5.7 Downloading Data from a DIS Stream

Function

This API is used to download data from a DIS stream.

When downloading data, you need to specify a data cursor. For details about how to obtain a cursor, see **Obtaining a Cursor**.

URI

- URI format

  GET /v2/{project_id}/records?{partition-cursor}
- Parameter description

  None

Request

- Example request

```
GET https://{endpoint}/v2/{project_id}/records?app_name=app1&partition-cursor=
eyJpdGVyR2VuVGltZSI6MTQ5MDk0ODk5OTM5NywiU3RyZWFtTmFtZSI6IjY2MCIsIlNoYXJkSWQiOiIwIiwiU2hhcmRJdGVyYXRvclR5cGUiOiJBVF9TRVFVRU5DRV9OVU1CRVIiLCJTdGFydGluZ1NlcXVlbmNlTnVtYmVyIjoiMCIsIlRpbWVTdGFtcCI6MH0=
```

- Parameter description

Table 5-14 Parameter description

Parameter	Mandatory	Type	Description
partition-cursor	Yes	String	Cursor of the partition used to specify the position in the partition from which to start reading data records sequentially. Value: 1 to 512 characters **NOTE** The validity period of a cursor is 5 minutes.
app_name	No	String	Name of the application that submits the checkpoint operation in the consumption stream.

Response

- Example response

```
{
  "records": [
    {
      "partition_key": "0",
      "sequence_number": "21",
       "data": "MTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTEx",
      "timestamp": 1527577402541,
      "timestamp_type": "CreateTime"
    }
  ],
  "next_partition_cursor":
"eyJpdGVyR2VuVGltZSI6MTQ5MDk1MDE1Nzc0NywiU3RyZWFtTmFtZSI6IjY2MCIsIlNoYXJkSWQiOiIwliwiU2hhcmRJdGVyYXRvclR5cGUiOiJBVF9TRVFVRU5DRV9OVU1CRVIiLCJTdGFydGluZ1NlcXVlbmNlTnVtYmVyIjoiMjIiLCJUaW1lU3RhbXAiOjB9",
 }
```

- Parameter description

Table 5-15 Response parameter description

Parameter	Type	Description
records	List<PullRecord>	Information of data records. For more information, see **Table 5-16**.
next_partition_cursor	String	Next cursor, which specifies the next position in the partition from which to start reading data records sequentially. **NOTE** The validity period of a cursor is 5 minutes.

Table 5-16 PullRecord parameter description

Parameter	Type	Description
data	String	Data pulled from the DIS stream.
partition_key	String	Partition key set when data is being uploaded. **NOTE** If the **partition_key** parameter is transferred when data is uploaded, this parameter is returned when data is downloaded. If the **partition_key** parameter is not transferred but the **partition_id** parameter is transferred when data is uploaded, no partition_key is returned.
sequence_number	String	Sequence number of an individual data record. Each data record has a sequence number that is unique within its partition. The sequence number is assigned by DIS when a data producer calls PutRecords to add data to a DIS stream. Sequence numbers for the same partition key generally increase over time; the longer the time period between write requests (PutRecords requests), the larger the sequence numbers become.
timestamp	Long	Timestamp when the record is written to DIS.
timestamp_type	String	Type of the timestamp. The value is **CreateTime**, specifying the creation time.

NOTE

- The data obtained through this API is the data encoded from the raw data using Base64 encoding.
- If a user downloads data using SDK provided by DIS, the SDK will automatically decode the data using Base64 decoding, and the user will obtain the raw data.

Status Code

- Normal

  200 OK
- Failed

  For more information, see **Error Codes**.

5.8 Creating an Application

Function

This API is used to create an application.

- Application: Multiple applications can access data in the same stream. Checkpoints generated for each application are used to record the consumed data in the stream by each application.
- An application and checkpoint are used together. When adding a checkpoint, you need to specify an application name. For details about how to add a checkpoint, see **Adding a Checkpoint**.

  Checkpoint: When an application consumes data, the latest SN of the consumed data is recorded as a checkpoint. When the data is reconsumed, the consumption can be continued based on this checkpoint.

URL

- URI format

  POST /v2/{project_id}/apps
- Parameter description

  None

Request

- Example request

```
POST https://{endpoint}/v2/{project_id}/apps

{
  "app_name": "app_test"
}
```

- Parameter description

  Table 5-17 Parameter description

Parameter	Mandatory	Type	Description
app_name	Yes	String	Unique name of the consumer application to be created. An application name is 1 to 200 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Response

- If the DIS stream was successfully created, a 201 response with an empty response body is returned.

- If the DIS stream failed to be created, identify the failure cause according to the response body and the instructions in **Error Codes**.

Status Code

- Normal

  201 Created
- Failed

  For more information, see **Error Codes**.

5.9 Deleting an Application

Function

This API is used to delete an application.

When deleting an application, you need to specify the application name. For details about the application, see **Viewing Details About a Consumer Application**.

URI

- URI format

  DELETE /v2/{project_id}/apps/{app_name}
- Parameter description

  None

Request

- Example request

```
DELETE
https://{endpoint}/v2/{project_id}/apps/apptest
```

- Parameter description

  None

Response

- If the DIS stream was successfully deleted, a 204 response with an empty response body is returned. For details about status codes, see **Status Codes**.
- If the DIS stream failed to be created, identify the failure cause according to the response body and the instructions in **Error Codes**.

Status Code

- Normal

  204 NOCONTENT
- Failed

  For more information, see **Error Codes**.

5.10 Viewing Details About a Consumer Application

Function

This API is used to query application details.

When querying application details, you need to specify the application name. The application name must be the same as that created in **Creating an Application**.

URI

- URI format

  GET /v2/{project_id}/apps/{app_name}

- Parameter description

  None

Request

- Example request

  ```
  GET https://{endpoint}/v2/{project_id}/apps/app_test
  ```

- Parameter description

  None

Response

- Example response

  ```
  {
  
        "app_id": "LGxEy8EvtdgzM7mSJfJ",
        "app_name": "app_test",
        "create_time": 1522313108126
    }
  ```

Table 5-18 Parameter description

Parameter	Type	Description
app_id	String	ID of the consumer application.
app_name	String	Name of the consumer application.
create_time	Long	Time when the consumer application is created. Unit: ms

Status Code

- Normal

  200 OK

- Failed

  For more information, see **Error Codes**.

5.11 Querying an Application List

Function

This API is used to query an application list.

During query, you need to specify the application from which the application list is returned and the maximum number of applications returned in a single request.

URI

- URI format

  GET /v2/{project_id}/apps
- Parameter description

  None

Request

- Example request

```
GET https://{endpoint}/v2/{project_id}/apps?limit=10&start_app_name=app_test
```

- Parameter description

Table 5-19 Parameter description

Parameter	Mandatory	Type	Description
limit	No	Int	Maximum number of applications to list in a single API call. Value range: 1 to 100 Default value: 10
start_app_name	No	String	Name of the application to start the list with. The returned application list does not contain this application name.
stream_name	No	String	Name of the stream whose application list will be returned.

Response

- Example response

```
{
    "has_more_app": false,
    "apps": [
        {

            "app_id": "LGxEy8EvtdgzM7mSJfJ",
```



```
        "app_name": "app",
        "create_time": 1522313108126
    },
    {

        "app_id": "xzRSZhpRN1x0BBgT0fR",
        "app_name": "dis-consumer-example",
        "create_time": 1532760945216
    }
  ]
}
```

Table 5-20 Response parameter description

Parameter	Type	Description
has_more_app	Boolean	Specifies whether there are more matching consumer applications to list. • **true**: There are more consumer applications. • **false**: There are no more consumer applications.
apps	List<App>	AppEntry list that meets the current request.

Table 5-21 Application parameter description

Parameter	Type	Description
app_id	String	ID of the consumer application.
app_name	String	Name of the consumer application.
create_time	Long	Time when the consumer application is created. Unit: ms

Status Code

- Normal
  200 OK
- Failed
  For more information, see **Error Codes**.

5.12 Adding a Checkpoint

Function

This API is used to add a checkpoint.

- When an application consumes data, the latest SN of the consumed data is recorded as a checkpoint. When the data is reconsumed, the consumption can be continued based on this checkpoint.
- An application and checkpoint are used together. When adding a checkpoint, you need to specify the stream name, partition ID, and application name.

URI

- URI format

  POST /v2/{project_id}/checkpoints
- Parameter description

  None

Request

- Example request

```
POST https://{endpoint}/v2/{project_id}/checkpoints

{
 "stream_name": "stream_name_test1",
  "app_name": "app_name1",
  "partition_id": "shardId-0000000000",
  "sequence_number": "10",
  "metadata": "metadata",
  "checkpoint_type": "LAST_READ"
}
```

- Parameter description

  Table 5-22 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream whose data record will have a checkpoint. A stream name is 1 to 64 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Mandatory	Type	Description
app_name	Yes	String	Unique name of the consumer application that will read data from the chosen DIS stream. An application name must contain 1 to 50 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
partition_id	Yes	String	Unique identifier of the partition.
sequence_number	Yes	String	Sequence number used to record the consumption checkpoint of the stream.
metadata	No	String	Metadata of the consumer application. The maximum metadata length is 1000 characters.
checkpoint_type	Yes	String	Type of the checkpoint. The checkpoint type **LAST_READ** indicates that only sequence numbers are recorded into the database.

Response

- If the DIS stream was successfully created, a 201 response with an empty response body is returned.
- If the DIS stream failed to be created, identify the failure cause according to the response body and the instructions in **Error Codes**.

Status Code

- Normal

  201 Created
- Failed

  For more information, see **Error Codes**.

5.13 Querying a Checkpoint

Function

This API is used to query a checkpoint.

When querying a checkpoint, you need to specify the stream name, partition ID, and application name.

URI

- URI format

  GET /v2/{project_id}/checkpoints{?stream_name,partition_id,app_name,checkpoint_type}

- Parameter description

  None

Request

- Example request

```
GET https://{ip}{endpoint}/v2/{project_id}/checkpoints?stream_name=stream_name_test&partition_id=0&app_name=app_name&checkpoint_type=LAST_READ
```

- Parameter description

  Table 5-23 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the stream created on the management console. A stream name is 1 to 64 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.
partition_id	Yes	String	Unique identifier of the partition.
app_name	Yes	String	Unique ID of the consumer application. An application name is 1 to 50 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.
checkpoint_type	Yes	String	Type of the checkpoint. The checkpoint type **LAST_READ** indicates that only sequence numbers are recorded into the database.

Response

- Example response

```
{
  "sequence_number": "10",
  "metadata": "metadata"
}
```

- Parameter description

Table 5-24 Response parameter description

Parameter	Type	Description
sequence_number	String	Unique sequence number. Each data record has a sequence number that is unique within its partition.
metadata	String	Metadata of the consumer application.

NOTE

If the checkpoint does not exist or expires, the value of **sequence_number** is **-1** and the value of **metadata** is empty.

Status Code

- Normal

  200 OK
- Failed

  For more information, see **Error Codes**.

5.14 Deleting a Checkpoint

Function

This API is used to delete a checkpoint.

When deleting a checkpoint, you need to specify the stream name, partition ID, and application name.

URI

- URI format

  DELETE /v2/{project_id}/checkpoints
- Parameter description

  None

Request

- Example request

  ```
  DELETE https://{endpoint}/v2/{project_id}/ checkpoints?stream_name=test&app_name=app
  ```

  – Parameter description

Table 5-25 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the stream created on the management console. A stream name is 1 to 64 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.
partition_id	No	String	Unique identifier of the partition.
app_name	Yes	String	Unique ID of the consumer application. An application name is 1 to 50 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.
checkpoint_type	No	String	Type of the checkpoint. **LAST_READ** indicates that only sequence numbers are recorded into the database.

Response

- If the DIS stream was successfully deleted, a 204 response with an empty response body is returned. For details about status codes, see **Status Codes**.
- If the DIS stream failed to be created, identify the failure cause according to the response body and the instructions in **Error Codes**.

Status Code

- Normal

  204 NOCONTENT
- Failed

  For more information, see **Error Codes**.

5.15 Changing Partition Quantity

Function

This API is used to change the number of partitions in a specific stream.

When changing the number of partitions, you need to specify the stream name and the number of target partitions.

URI

- URI format

  PUT /v2/{project_id}/streams/{stream_name}

- Parameter description

  None

Request

- Example request

```
PUT https://{endpoint}/v2/{project_id}/streams/stream_name_test

{
"stream_name":"stream_name_test",
"target_partition_count":5
}
```

- Parameter description

  Table 5-26 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the stream created on the management console. A stream name is 1 to 64 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.
target_partition_count	Yes	int	Number of the target partitions. The value is an integer larger than 0. If the value is greater than the number of the current partitions, scale-up will be performed. If the value is smaller than the number of the current partitions, scale-down will be performed. **NOTE** Each stream can be scaled up and down a total of five times within one hour. After the stream is successfully scaled up or down, it cannot be scaled up or down again within the next one hour.

Response

- Example response

```
{
"stream_name":"stream_name_test",
"current_partition_count":2
"target_partition_count":5
}
```


- Parameter description

  Table 5-27 Response parameter description

Parameter	Type	Description
stream_name	String	Name of the stream whose partition quantity needs to be changed.
current_partition_count	int	Number of the current partitions
target_partition_count	int	Number of the target partitions.

Status Code

- Normal

  200 OK
- Failed

  For more information, see **Error Codes**.

5.16 Querying Stream Monitoring Information

Function

This API is used to query stream monitoring information.

When querying the stream monitoring information, you need to specify the stream name, monitoring index, and start time and end time of the monitoring.

URI

- URI format

  GET /v2/{project_id}/streams/{stream_name}/metrics?label={label}&label_list={label_list}&start_time={start_time}&end_time={end_time}
- Parameter description

  None

Request

- Example request

  Label request:

```
GET https://{endpoint}/v2/{project_id}/streams/stream_test/metrics?label=total_put_records_per_stream&start_time=1534917441&end_time=1534921041
```

  label_list request:

```
GET https://{endpoint}/v2/{project_id}/streams/stream_test/metrics?label_list=total_put_bytes_per_stream,total_put_records_per_stream&start_time=1534917441&end_time=1534921041
```

- Parameter description

Table 5-28 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream to be queried.

Parameter	Mandatory	Type	Description
label	No	String	Stream monitoring metric. (Either **label** or **label_list** must be set. If both **label_list** and **label** exist, **label_list** prevails.) • total_put_bytes_per_stream: total input traffic (byte) • total_get_bytes_per_stream: total output traffic (byte) • total_put_records_per_stream: total number of input records • total_get_records_per_stream: total number of output records • total_put_req_latency: average processing time of upload requests (ms) • total_get_req_latency: average processing time of download requests (ms) • total_put_req_suc_per_stream: number of successful upload requests

Parameter	Mandatory	Type	Description
			• Total_get_req_suc_per_stream: number of successful download requests • traffic_control_put: number of rejected upload requests due to flow control • traffic_control_get: number of rejected download requests due to flow control
label_list	No	String	List of labels separated by commas (,) to query multiple labels in batches. (Either **label** or **label_list** must be set. If both **label_list** and **label** exist, **label_list** prevails.)
start_time	Yes	Long	Monitoring start time, which is a 10-digit timestamp.
end_time	Yes	Long	Monitoring end time, which is a 10-digit timestamp.

Response

- Example response

  Label query

```
{
  "metrics": {
```

```
  "dataPoints": [
  {
    "timestamp": 1533200460,
    "value": 2760
  },
  {
    "timestamp": 1533200340,
    "value": 3575
  }
  ],
  "label": "total_put_records_per_stream"
 }
}
```

label_list query

```
{
   "metrics_list": [{
       "label": "total_put_bytes_per_stream",
       "dataPoints": [{
           "timestamp": 1575537840,
           "value": 81111040
       }, {
           "timestamp": 1575536640,
           "value": 92160000
       }]
   }, {
       "label": "total_put_records_per_stream",
       "dataPoints": [{
           "timestamp": 1575537840,
           "value": 7921
       }, {
           "timestamp": 1575536640,
           "value": 9000
       }]
   }]
}
```

- Parameter description

Table 5-29 Response parameter description

Parameter	Type	Description
metrics	Metrics	Monitoring data object.
metrics_list	List<Metrics>	List of monitored data objects.

Table 5-30 Metrics parameter description

Parameter	Type	Description
dataPoints	List<DataPoint>	Stream monitoring data.

Parameter	Type	Description
label	String	Stream monitoring metrics. • total_put_bytes_per_stream: total input traffic (byte) • total_get_bytes_per_stream: total output traffic (byte) • total_put_records_per_stream: total number of input records • total_get_records_per_stream: total number of output records • total_put_req_latency: average processing time of upload requests (ms) • total_get_req_latency: average processing time of download requests (ms) • total_put_req_suc_per_stream: number of successful upload requests • Total_get_req_suc_per_stream: number of successful download requests • traffic_controll_put: number of rejected upload requests due to flow control • traffic_controll_get: number of rejected download requests due to flow control

Table 5-31 DataPoint parameter description

Parameter	Type	Description
timestamp	Long	Specifies the timestamp.
value	Long	Specifies the monitoring value corresponding to the timestamp.

Status Code

- Normal

  200 OK
- Failed

  For more information, see **Error Codes**.

5.17 Querying Partition Monitoring Information

Function

This API is used to query partition monitoring information.

When querying the partition monitoring information, you need to specify the stream name, partition ID, monitoring index, and start time and end time of the monitoring.

URI

- URI format

  GET /v2/{project_id}/streams/{stream_name}/partitions/{partition_id}/metrics?label={label}&start_time={start_time}&end_time={end_time}
- Parameter description

  None

Request

- Example request

  ```
  GET https://{endpoint}/v2/{project_id}/streams/stream_test/partitions/shardId-0000000000/metrics?label=total_put_bytes_per_partition&start_time=1534917441&end_time=1534921041
  ```
- Parameter description

Table 5-32 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream to be queried.
partition_id	Yes	String	Unique identifier of the partition.
label	Yes	String	Partition monitoring metrics. • total_put_bytes_per_partition: total input traffic in a partition (byte) • total_get_bytes_per_partition: total output traffic in a partition (byte) • total_put_records_per_partition: total number of input records in a partition • total_get_records_per_partition: total number of output records in a partition
start_time	Yes	Long	Monitoring start time, which is a 10-digit timestamp.
end_time	Yes	Long	Monitoring end time, which is a 10-digit timestamp.

Response

- Example response

```
{
  "metrics": {
    "dataPoints": [
    {
      "timestamp": 1533200460,
      "value": 14130
    },
    {
      "timestamp": 1533200340,
      "value": 18030
    }
    ],
    "label": "total_put_bytes_per_partition"
  }
}
```

- Parameter description

Table 5-33 Response parameter description

Parameter	Type	Description
metrics	Object	Monitoring data object.

Table 5-34 metrics parameter description

Parameter	Type	Description
dataPoints	List<DataPoint>	Partition monitoring data.
label	String	Partition monitoring metrics. • total_put_bytes_per_partition: total input traffic in a partition (byte) • total_get_bytes_per_partition: total output traffic in a partition (byte) • total_put_records_per_partition: total number of input records in a partition • total_get_records_per_partition: total number of output records in a partition

Table 5-35 DataPoint parameter description

Parameter	Type	Description
timestamp	Long	Specifies the timestamp.
value	Long	Specifies the monitoring value corresponding to the timestamp.

Status Code

- Normal
  200 OK

- Failed
  For more information, see **Error Codes**.

5.18 Adding a Dump Task

Function

This API is used to add a dump task.

When adding a dump task, you need to specify the stream name.

URI

- URI format
  POST /v2/{project_id}/streams/{stream_name}/transfer-tasks
- Parameter description
  None

Request

- Example request
  Add an OBS dump task. The dump file format is Text.

```
POST https://{endpoint}/v2/{project_id}/streams/{stream_name}/transfer-tasks
{
   "destination_type": "OBS",
   "obs_destination_descriptor": {
      "task_name": "task_DtDO",
      "consumer_strategy": "LATEST",
      "agency_name": "dis-global",
      "destination_file_type": "text",
      "obs_bucket_path": "111111111",
      "file_prefix": "0823",
      "partition_format": "yyyy/MM/dd/HH/mm",
      "record_delimiter": "|",
      "deliver_time_interval": 30
   }
}
```

  Add an OBS dump task. The dump file format is Parquet.

```
POST https://{endpoint}/v2/{project_id}/streams/{stream_name}/transfer-tasks
{
   "destination_type": "OBS",
   "obs_destination_descriptor": {
      "task_name": "task_DtDO",
      "consumer_strategy": "LATEST",
      "agency_name": "dis-global",
      "destination_file_type": "parquet",
      "obs_bucket_path": "111111111",
      "file_prefix": "0823",
      "partition_format": "yyyy/MM/dd/HH/mm",
      "record_delimiter": "|",
      "deliver_time_interval": 30,
      "processing_schema": {
         "timestamp_name": "date",
         "timestamp_type": "String",
         "timestamp_format": "yyyy/MM/dd HH:mm:ss"
      }
   }
}
```


Add an OBS dump task. The dump file format is CarbonData.

```
POST https://{endpoint}/v2/{project_id}/streams/{stream_name}/transfer-tasks

{
  "destination_type": "OBS",
   "obs_destination_descriptor":
     {
        "task_name":"task_DtDO",
         "consumer_strategy":"LATEST",
         "agency_name":"dis-global",
"destination_file_type":"CarbonData",
         "obs_bucket_path":"111111111",
         "file_prefix":"0823",
         "partition_format":"yyyy/MM/dd/HH/mm",
         "record_delimiter":"|",
         "deliver_time_interval":30
     }
 }
```

Add an MRS dump task. The dump file format is Text.

```
POST https://{endpoint}/v2/{project_id}/streams/{stream_name}/transfer-tasks

 {
  "destination_type": "MRS",
   "mrs_destination_descriptor":{
        "task_name":"task_8CWn",
        "consumer_strategy":"TRIM_HORIZON",
        "agency_name":"dis-global",
        "destination_file_type":"text",
        "mrs_cluster_id":"f8123fa6-99f1-4ed9-83f4-c827c7277d41",
        "mrs_cluster_name":"mrs_jxZP",
        "mrs_hdfs_path":"/user",
        "obs_bucket_path":"111111111",
        "file_prefix":"test",
        "hdfs_prefix_folder":"test",
        "deliver_time_interval":30,
        "retry_duration":1800
    }
}
```

Add an MRS dump task. The dump file format is Parquet.

```
POST https://{endpoint}/v2/{project_id}/streams/{stream_name}/transfer-tasks

 {
  "destination_type": "MRS",
   "mrs_destination_descriptor":{
        "task_name":"task_JvzV",
        "consumer_strategy":"LATEST",
        "agency_name":"dis-global",
        "destination_file_type":"parquet",
        "mrs_cluster_id":"13cfefa4-a410-4717-91ba-1b694ec92da7",
        "mrs_cluster_name":"mrs_gkFP_172",
        "mrs_hdfs_path":"/user",
        "obs_bucket_path":"111111111",
        "file_prefix":"test",
        "hdfs_prefix_folder":"test",
        "deliver_time_interval":30,
        "retry_duration":1800
    }
}
```

Add an MRS dump task. The dump file format is CarbonData.

```
POST https://{endpoint}/v2/{project_id}/streams/{stream_name}/transfer-tasks

 {
  "destination_type": "MRS",
   "mrs_destination_descriptor":{
        "task_name":"task_w6WG",
```



```
        "consumer_strategy":"TRIM_HORIZON",
        "agency_name":"dis-global",
"destination_file_type":"CarbonData",
        "carbon_properties":null,
        "mrs_cluster_id":"13cfefa4-a410-4717-91ba-1b694ec92da7",
        "mrs_cluster_name":"mrs_gkFP_172",
        "mrs_hdfs_path":"/user",
        "obs_bucket_path":"111111111",
        "file_prefix":"test",
        "hdfs_prefix_folder":"test",
        "deliver_time_interval":30,
        "retry_duration":1800
    }
}
```

Add a DLI dump task.

```
POST https://{endpoint}/v2/{project_id}/streams/{stream_name}/transfer-tasks
{
    "destination_type": "DLI",
    "dli_destination_descriptor": {
        "task_name": "task",
        "consumer_strategy": "LATEST",
        "agency_name": "dli-agency",
        "dli_database_name": "db",
        "dli_table_name": "table",
        "obs_bucket_path": "1111",
        "file_prefix": "2018",
        "deliver_time_interval": 300,
        "retry_duration ": 300
    }
}
```

Add a CloudTable HBase dump task.

```
POST https://{endpoint}/v2/{project_id}/streams/{stream_name}/transfer-tasks
{
    "destination_type": "CLOUDTABLE",
    "cloudtable_destination_descriptor": {
        "task_name": "task",
        "consumer_strategy": "TRIM_HORIZON",
        "cloudtable_cluster_name": "cloudtable_cluster",
        "cloudtable_cluster_id": "b8c095e2-db5f-4732-8a1d-eacd662e35dc",
        "cloudtable_table_name": "cloudtable_table",
        "cloudtable_row_key_delimiter": "|",
        "retry_duration": 1800,
        "obs_backup_bucket_path": "obs-test-hz",
        "backup_file_prefix": "2018",
        "cloudtable_schema": {
            "row_key": [{
                "value": "dataId",
                "type": "String"
            }],
            "columns": [{
                "column_family_name": "cfname1",
                "column_name": "ID",
                "value": "dataId",
                "type": "String"
            }, {
                "column_family_name": "cfname2",
                "column_name": "VALUE",
                "value": "dataValue",
                "type": "String"
            }]
        }
    }
}
```

Add a CloudTable OpenTSDB dump task.

```
{
    "destination_type": "CLOUDTABLE",
```



```
  "cloudtable_destination_descriptor": {
     "task_name": "task",
     "consumer_strategy": "LATEST",
     "cloudtable_cluster_name": "cloudtable_cluster",
     "cloudtable_cluster_id": "b8c095e2-db5f-4732-8a1d-eacd662e35dc",
     "retry_duration": 1800,
     "obs_backup_bucket_path": "obs-test-hz",
     "backup_file_prefix": "2018",
     "opentsdb_schema": [{
        "metric": [{
           "type": "Constant",
           "value": "age"
        }],
        "timestamp": {
           "value": "date",
           "type": "String",
           "format": "yyyy/MM/dd HH:mm:ss"
        },
        "value": {
           "value": "value",
           "type": "Bigint"
        },
        "tags": [{
           "name": "name",
           "value": "name",
           "type": "Bigint"
        }]
     }]
  }
}
```

Add a DWS dump task.

```
{
  "destination_type": "DWS",
       "dws_destination_descriptor":              {
     "task_name": "task",
     "consumer_strategy": "LATEST",
                    "agency_name": "dis-global",
                    "dws_cluster_name": "dws-Shang",
                    "dws_cluster_id": "ea4a0a58-7b02-4d56-a3ec-ed67b498f8b9",
                    "dws_database_name": "postgres",
                    "dws_schema": "dbadmin",
                    "dws_table_name": "dws_table",
                    "dws_delimiter": "|",
                    "user_name": "dbadmin",
                    "user_password": "password",
                    "kms_user_key_name": "KMS-key",
                    "kms_user_key_id": "2260a755-c256-4211-aace-3c4358099b08",
                    "obs_bucket_path": "dws-test",
                    "file_prefix": "a",
                    "deliver_time_interval": 300,
                    "retry_duration": 1800,
     "dws_table_columns": "dev,id,online,module,appId,logTime,evvvv",
     "options": {
        "fill_missing_fields": "false",
        "ignore_extra_data": "false",
        "compatible_illegal_chars": "false"
     }
  }
}
```


Table 5-36 Parameter description

Parameter	Mandatory	Type	Description
destination_type	No	String	Dump destination. Possible values: • **OBS** • **MRS** • **DLI** • **CLOUDTABLE** • **DWS**
obs_destination_descriptor	No	Object	Parameter list of the OBS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to OBS.
mrs_destination_descriptor	No	Object	Parameter list of the MRS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to MRS.
dli_destination_descriptor	No	Object	Parameter list of the DLI to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to DLI.
cloudtable_destination_descriptor	No	Object	Parameter list of the CloudTable to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to CloudTable HBase or OpenTSDB.

Parameter	Mandatory	Type	Description
dws_destination_descriptor	No	Object	Parameter list of the DWS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to DWS.

Table 5-37 obs_destination_descriptor parameter description

Parameter	Mandatory	Type	Description
task_name	Yes	String	Name of the dump task. A task name is 1 to 64 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.
destination_file_type	No	String	Dump file format. Possible values: • **Text** • **Parquet** • **CarbonData** **NOTE** • The **Parquet** or **CarbonData** format can be selected only when **Source Data Type** is set to **JSON** and **Dump Destination** is set to **OBS**. • Default value: text
agency_name	Yes	String	Name of the agency created in IAM. DIS uses an agency to access your specified resources. Agency parameter settings: • **Agency Type**: **Cloud service** • **Cloud Service**: **DIS** • **Validity Period**: **Permanent** • Set **Policy** to **Tenant Administrator** on the OBS project in the **Global** service region. This parameter cannot be left unspecified and the parameter value cannot exceed 64 characters.

Parameter	Mandatory	Type	Description
obs_bucket_path	Yes	String	Name of the OBS bucket used to store data from the DIS stream.
file_prefix	No	String	Directory to store files that will be dumped to OBS. Different directory levels are separated by slashes (/) and cannot start with slashes. **NOTE** This parameter is optional if streaming data is dumped to OBS. The value cannot exceed 50 characters. Only letters, digits, underscores (_), and slashes (/) are allowed. This parameter is left unspecified by default.
consumer_strategy	No	String	Specifies the offset. • **Latest**: Maximum offset, indicating that the latest data will be extracted. • **TRIM_HORIZON**: Minimum offset, indicating that the earliest data will be extracted. Default value: Latest
deliver_time_interval	Yes	Int	User-defined interval at which data is imported from the current DIS stream into OBS. If no data is pushed to the DIS stream during the current interval, no dump file package will be generated. **NOTE** This parameter is mandatory if streaming data is dumped to OBS. Value range: 30s to 900s Default value: 300 Unit: second

Parameter	Mandatory	Type	Description
partition_format	No	String	Directory structure of the object file written into OBS. The directory structure is in the format of yyyy/MM/dd/HH/mm (time at which the dump task was created). • N/A: If this parameter is left unspecified, the time directory format will not be used. • yyyy: year. • yyyy/MM: year and month. • yyyy/MM/dd: year, month, and day. • yyyy/MM/dd/HH: year, month, day, and hour. • yyyy/MM/dd/HH/mm: year, month, day, hour, and minute. For example, if the dump task was created at 14:49:00 on November 10, 2017, then the directory structure is **2017/11/10/14/49**. **2017** is the level-1 directory. This parameter is left unspecified by default. **NOTE** After the data is dumped successfully, the directory structure is **obs_bucket_path/file_prefix/partition_format**.
processing_schema	No	Object	Directory structure of the object file written into OBS. The directory structure is in the format of yyyy/MM/dd/HH/mm and generated based on the source data timestamp and the configured **partition_format**. **NOTE** This parameter is mandatory if you want to customize the OBS directory to which the parquet file will be dumped.

Parameter	Mandatory	Type	Description
record_delimiter	No	String	Delimiter for the dump file, which is used to separate the user data that is written into the dump file. Possible values: • Comma (,) • Semicolon (;) • Vertical bar (|) • Newline (\n) • NULL Default value: \n

Table 5-38 processing_schema parameter description

Parameter	Mandatory	Type	Description
timestamp_name	Yes	String	Attribute name of the source data timestamp.
timestamp_type	Yes	String	Type of the source data timestamp. • String • Timestamp: 13-bit timestamp of the long type
timestamp_format	No	String	OBS directory generated based on the timestamp format. This parameter is mandatory when the timestamp type of the source data is String. Possible values: • yyyy/MM/dd HH:mm:ss • MM/dd/yyyy HH:mm:ss • dd/MM/yyyy HH:mm:ss • yyyy-MM-dd HH:mm:ss • MM-dd-yyyy HH:mm:ss • dd-MM-yyyy HH:mm:ss

NOTE

The data dumped to OBS is the raw data uploaded by users, that is, the data decoded using Base64.

Table 5-39 mrs_destination_descriptor parameter description

Parameter	Mandatory	Type	Description
agency_name	Yes	String	Name of the agency created in IAM. DIS uses an agency to access your specified resources. Agency parameter settings: • **Agency Type**: **Cloud service** • **Cloud Service**: **DIS** • **Validity Period**: **Permanent** • **Policy** **NOTE** Both of the following permissions need to be configured. – Set **Policy** to **Tenant Administrator** on the OBS project in the **Global** service region. – Set **Policy** to **MRS Administrator**, **Server Administrator**, and **Tenant Guest** on the project in the region to which the chosen MRS cluster belongs. This parameter cannot be left unspecified and the parameter value cannot exceed 64 characters.
mrs_cluster_name	Yes	String	Name of the MRS cluster to which data in the DIS stream will be dumped. **NOTE** Only the MRS cluster authenticated not by Kerberos is supported.
mrs_cluster_id	Yes	String	ID of the MRS cluster to which data in the DIS stream will be dumped.
mrs_hdfs_path	Yes	String	Hadoop Distributed File System (HDFS) path of the MRS cluster to which data in the DIS stream will be dumped.
obs_bucket_path	Yes	String	Name of the OBS bucket to which data in the DIS stream will be temporarily saved.

Parameter	Mandatory	Type	Description
file_prefix	No	String	Self-defined directory created in the OBS bucket and used to temporarily store data in the DIS stream. Directory levels are separated by slashes (/) and cannot start with slashes. The value cannot exceed 50 characters. Only letters, digits, underscores (_), and slashes (/) are allowed. This parameter is left unspecified by default.
hdfs_prefix_folder	Yes	string	Directory to store files that will be dumped to HDFS of the chosen MRS cluster. Different directory levels are separated by slashes (/). This directory name is 0 to 50 characters long. This parameter is left unspecified by default.
consumer_strategy	No	String	Specifies the offset. • **Latest**: Maximum offset, indicating that the latest data will be extracted. • **TRIM_HORIZON**: Minimum offset, indicating that the earliest data will be extracted. Default value: Latest
deliver_time_interval	Yes	Int	User-defined interval at which data is imported from the DIS stream into HDFS of the chosen MRS cluster. If no data is pushed to the DIS stream during the current interval, no dump file package will be generated. Value range: 30s to 900s Unit: second Default value: 300

Parameter	Mandatory	Type	Description
retry_duration	No	Int	Time duration for DIS to retry if data fails to be dumped. If the duration is exceeded but the dump still fails, the data will be backed up to *OBS bucket name*/*file_prefix*/**mrs_error**. Value range: 0s to 7200s Unit: second Default value: 1800 If the value is set to **0**, no retry is allowed.

Table 5-40 dli_destination_descriptor parameter description

Parameter	Mandatory	Type	Description
agency_name	Yes	String	Name of the agency created in IAM. DIS uses an agency to access your specified resources. Agency parameter settings: • **Agency Type**: **Cloud service** • **Cloud Service**: **DIS** • **Validity Period**: **Permanent** • **Policy** **NOTE** Both of the following permissions need to be configured. • Set **Policy** to **Tenant Administrator** on the OBS project in the **Global** service region. • Set **Policy** to the **DLI Service User** permission on the project in the region to which the chosen DLI table belongs. This parameter cannot be left unspecified and the parameter value cannot exceed 64 characters.
dli_database_name	Yes	String	Name of the DLI database to which data in the DIS stream will be dumped.

Parameter	Mandatory	Type	Description
dli_table_name	Yes	String	Name of the DLI table to which data in the DIS stream will be dumped. **NOTE** Currently, only DLI tables are supported. Before selecting a DLI table, ensure that you have the permission to insert a DLI table.
consumer_strategy	No	String	Specifies the offset. • **Latest**: Maximum offset, indicating that the latest data will be extracted. • **TRIM_HORIZON**: Minimum offset, indicating that the earliest data will be extracted. Default value: Latest
obs_bucket_path	Yes	String	Name of the OBS bucket to which data in the DIS stream will be temporarily saved.
file_prefix	No	String	Self-defined directory created in the OBS bucket and used to temporarily store data in the DIS stream. Directory levels are separated by slashes (/) and cannot start with slashes. The value cannot exceed 50 characters. Only letters, digits, underscores (_), and slashes (/) are allowed. This parameter is left unspecified by default.
deliver_time_interval	Yes	Int	User-defined interval at which data is imported from the DIS stream into DLI. If no data is pushed to the DIS stream during the current interval, no dump file package will be generated. Value range: 30s to 900s Unit: second Default value: 300

Parameter	Mandatory	Type	Description
retry_duration	No	Int	Time duration for DIS to retry if data fails to be dumped to DLI. If the duration is exceeded but the dump still fails, the data will be backed up to *OBS bucket name*/*file_prefix*/**dli_error**. Value range: 0s to 7200s Unit: second Default value: 1800 If the value is set to **0**, no retry is allowed.

Table 5-41 cloudtable_destination_descriptor parameter description

Parameter	Mandatory	Type	Description
consumer_strategy	No	String	Specifies the offset. • **Latest**: Maximum offset, indicating that the latest data will be extracted. • **TRIM_HORIZON**: Minimum offset, indicating that the earliest data will be extracted. Default value: Latest
cloudtable_cluster_name	Yes	String	Name of the CloudTable cluster to which data will be dumped. If data will be dumped to OpenTSDB, OpenTSDB must be enabled before the dump.
cloudtable_cluster_id	Yes	String	ID of the CloudTable cluster to which data will be dumped. If data will be dumped to OpenTSDB, OpenTSDB must be enabled before the dump.
cloudtable_table_name	No	String	HBase table name of the CloudTable cluster to which data will be dumped. The parameter is mandatory when data is dumped to the CloudTable HBase.

Parameter	Mandatory	Type	Description
cloudtable_schema	No	Object	Schema configuration of the CloudTable HBase data. You can set either this parameter or **opentsdb_schema**, but this parameter is mandatory when data will be dumped to HBase. After this parameter is set, the JSON data in the stream can be converted to another format and then be imported to the CloudTable HBase.
cloudtable_row_key_delimiter	No	String	Delimiter used to separate the user data that generates HBase row keys. Possible values: , . | ; \ - _ ~ Default value: .
opentsdb_schema	No	List<Object>	Schema configuration of the CloudTable OpenTSDB data. You can set either this parameter or **cloudtable_schema**, but this parameter is mandatory when data will be dumped to OpenTSDB. After this parameter is set, the JSON data in the stream can be converted to another format and then be imported to the CloudTable OpenTSDB.
obs_backup_bucket_path	No	String	Name of the OBS bucket used to back up data that failed to be dumped to CloudTable.
backup_file_prefix	No	String	Self-defined directory created in the OBS bucket and used to back up data that failed to be dumped to CloudTable. Directory levels are separated by slashes (/) and cannot start with slashes. A directory name cannot exceed 50 characters. Only letters, digits, and underscores (_) are allowed. This parameter is left unspecified by default.

Parameter	Mandatory	Type	Description
retry_duration	No	Int	Time duration for DIS to retry if data fails to be dumped to CloudTable. If the duration is exceeded but the dump still fails, the data will be backed up to *OBS bucket name*/*backup_file_prefix*/**cloudtable_error** or *OBS bucket name*/*backup_file_prefix*/**opentsdb_error**. Value range: 0s to 7200s Unit: second Default value: 1800

Table 5-42 cloudtable_schema parameter description

Parameter	Mandatory	Type	Description
row_key	Yes	List<Object>	HBase rowkey schema used by the CloudTable cluster to convert JSON data into HBase rowkeys. Value range: 1 to 64
columns	Yes	List<Object>	HBase column schema used by the CloudTable cluster to convert JSON data into HBase columns. Value range: 1 to 4096

Table 5-43 row_key parameter description

Parameter	Mandatory	Type	Description
value	Yes	String	JSON attribute name, which is used to generate HBase rowkeys for JSON data in the DIS stream.

Parameter	Mandatory	Type	Description
type	Yes	**String**	JSON attribute type of JSON data in the DIS stream. Possible values: • **Bigint** • **Double** • **Boolean** • **Timestamp** • **String** • **Decimal**

Table 5-44 columns parameter description

Parameter	Mandatory	Type	Description
column_family_name	Yes	String	Name of the HBase column family to which data will be dumped.
column_name	Yes	String	Name of the HBase column to which data will be dumped. This value is 1 to 32 characters long. Only letters, digits, and underscores (_) are allowed.
value	Yes	String	JSON attribute name, which is used to generate HBase column values for JSON data in the DIS stream.
type	Yes	String	JSON attribute type of JSON data in the DIS stream. Possible values: • **Bigint** • **Double** • **Boolean** • **Timestamp** • **String** • **Decimal**

Table 5-45 opentsdb_schema parameter description

Parameter	Mandatory	Type	Description
metric	Yes	List<Object>	Schema configuration of the OpenTSDB data metric in the CloudTable cluster. After this parameter is set, the JSON data in the stream can be converted to the metric of the OpenTSDB data. Value: 1
timestamp	Yes	Object	Schema configuration of the OpenTSDB data timestamp in the CloudTable cluster. After this parameter is set, the JSON data in the stream can be converted to the timestamp of the OpenTSDB data.
value	Yes	Object	Schema configuration of the OpenTSDB data value in the CloudTable cluster. After this parameter is set, the JSON data in the stream can be converted to the value of the OpenTSDB data.
tags	Yes	List<Object>	Schema configuration of the OpenTSDB data tags in the CloudTable cluster. After this parameter is set, the JSON data in the stream can be converted to the tags of the OpenTSDB data. Value range: 1 to 8

Table 5-46 metric parameter description

Parameter	Mandatory	Type	Description
type	Yes	String	• When **type** is set to **Constant**, the value of **metric** is the value of **value**. • When **value** is set to **String**, the value of **metric** is the value of the JSON attribute of the user data in the stream.

Parameter	Mandatory	Type	Description
value	Yes	String	Constant value or JSON attribute name of the user data in the stream. This value is 1 to 32 characters long. Only letters, digits, and periods (.) are allowed.

Table 5-47 timestamp parameter description

Parameter	Mandatory	Type	Description
type	Yes	String	• When **type** is set to **Timestamp**, the value type of the JSON attribute of the user data in the stream is **Timestamp**, and the timestamp of OpenTSDB can be generated without converting the data format. • When **type** is set to **String**, the value type of the JSON attribute of the user data in the stream is **Date**, and the timestamp of OpenTSDB can be generated only after the data format is converted.
value	Yes	String	JSON attribute name of the user data in the stream. This value is 1 to 32 characters long. Only letters, digits, and underscores (_) are allowed.
format	No	String	This parameter is mandatory when **type** is set to **String**. When the value type of the JSON attribute of the user data in the stream is **Date**, **format** is required to convert the data format to generate the timestamp of OpenTSDB. Possible values: • yyyy/MM/dd HH:mm:ss • MM/dd/yyyy HH:mm:ss • dd/MM/yyyy HH:mm:ss • yyyy-MM-dd HH:mm:ss • MM-dd-yyyy HH:mm:ss • dd-MM-yyyy HH:mm:ss

Table 5-48 value parameter description

Parameter	Mandatory	Type	Description
type	Yes	String	Type name of the JSON attribute of the user data in the stream. Possible values: • **Bigint** • **Double** • **Boolean** • **Timestamp** • **String** • **Decimal**
value	Yes	String	JSON attribute name of the user data in the stream. This value is 1 to 32 characters long. Only letters, digits, and periods (.) are allowed.

Table 5-49 tags parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Tag name of the OpenTSDB data that stores the data in the stream. This value is 1 to 32 characters long. Only letters, digits, and underscores (_) are allowed.
type	Yes	String	Type name of the JSON attribute of the user data in the stream. Possible values: • **Bigint** • **Double** • **Boolean** • **Timestamp** • **String** • **Decimal**
value	Yes	String	Constant value or JSON attribute name of the user data in the stream. This value is 1 to 32 characters long. Only letters, digits, and underscores (_) are allowed.

Table 5-50 dws_destination_descriptor parameter description

Parameter	Mandatory	Type	Description
consumer_strategy	No	String	Specifies the offset. • **Latest**: Maximum offset, indicating that the latest data will be extracted. • **TRIM_HORIZON**: Minimum offset, indicating that the earliest data will be extracted. Default value: Latest
agency_name	Yes	String	Name of the agency created in IAM. DIS uses an agency to access your specified resources. Agency parameter settings: • **Agency Type**: **Cloud service** • **Cloud Service**: **DIS** • **Validity Period**: **Permanent** • **Policy** **NOTE** Both of the following permissions need to be configured. – Set **Policy** to **Tenant Administrator** on the OBS project in the **Global** service region. – Set **Policy** to **DWS Administrator**, **Server Administrator**, and **Tenant Guest** on the project and region to which the DWS cluster belongs. This parameter cannot be left unspecified and the parameter value cannot exceed 64 characters.
dws_cluster_name	Yes	String	Name of the DWS cluster to which data will be dumped.
dws_cluster_id	Yes	String	ID of the DWS cluster to which will be dumped.
dws_database_name	Yes	String	Name of the DWS database to which data in the DIS stream will be dumped.
dws_schema	Yes	String	Schema of the DWS database to which data will be dumped.

Parameter	Mandatory	Type	Description
dws_table_name	Yes	String	Name of the table in the DWS database to which data will be dumped.
dws_delimiter	Yes	String	Delimiter used to separate the columns in the DWS tables. The delimiter can be a comma (**,**), semicolon (**;**), or vertical bar (**|**).
user_name	Yes	String	Username of the DWS database to which data will be dumped.
user_password	Yes	String	Password of the DWS database to which data will be dumped.
kms_user_key_name	Yes	String	Key created in Key Management Service (KMS) and used to encrypt the password of the DWS database.
kms_user_key_id	Yes	String	ID of the key created in KMS and used to encrypt the password of the DWS database.
obs_bucket_path	Yes	String	Name of the OBS bucket to which data in the DIS stream will be temporarily saved.
file_prefix	No	String	Self-defined directory created in the OBS bucket and used to temporarily store data in the DIS stream. Directory levels are separated by slashes (/) and cannot start with slashes. The value cannot exceed 50 characters. Only letters, digits, underscores (_), and slashes (/) are allowed. This parameter is left unspecified by default.
deliver_time_interval	Yes	Int	Interval at which data is imported to the table of a DWS cluster. If no data is imported during the specific period of time, no dump file package is created. Value range: 30s to 900s Unit: second Default value: 300

Parameter	Mandatory	Type	Description
retry_duration	No	Int	Time duration for DIS to retry if data failed to be dumped to the DWS cluster. If the duration is exceeded but the dump still fails, the data will be backed up to *OBS bucket name*/*file_prefix*/**dws_error**. Value range: 0s to 7200s Unit: second Default value: 1800
dws_table_columns	No	String	Column to be dumped to the DWS table. If the value is null or empty, all columns are dumped by default. The value is left blank by default.
options	No	Object	DWS fault tolerance option (used to specify various parameters of foreign table data).

Table 5-51 options parameter description

Parameter	Mandatory	Type	Description
fill_missing_fields	No	String	Specifies whether to set the field to Null or enable an error message to be displayed in the error table when the last field in a row of the data source file is missing during database import. Value: **true/on** or **false/off** Default value: **false/off**
ignore_extra_data	No	String	Specifies whether to ignore the extra columns when the number of fields in the data source file is greater than the number of columns defined in the foreign table. This parameter is used only during data import. Value: **true/on** or **false/off** Default value: **false/off**

Parameter	Mandatory	Type	Description
compatible_illegal_chars	No	String	Specifies whether to convert invalid characters based on the conversion rule and save the converted characters to the database, or whether to report an error and stop the import. Value: **true/on** or **false/off** Default value: **false/off**
reject_limit	No	String	Specifies the maximum number of data format errors allowed during the data import. If the number of data format errors does not reach the maximum, the data import is successful. Value range: integer or unlimited (no limit) Default value: 0, indicating that an error message is returned immediately a data format error occurs.

Response

- If the operation is successfully performed, status code 201 is returned and the response message body is empty.
- If the operation failed to be performed, identify the failure cause according to the response body and the instructions in **Error Codes**.

Response Code

- Normal

  201 Created
- Failed

  For more information, see **Error Codes**.

5.19 Deleting a Dump Task

Function

This API is used to delete a dump task.

When deleting a dump task, you need to specify the stream name and dump task name.

URI

- URI format

  DELETE /v2/{project_id}/streams/{stream_name}/transfer-tasks/{transfer_task_name}
- Parameter description

  None

Request

- Example request

  ```
  DELETE https://{endpoint}/v2/{project_id}/streams/stream_test/transfer-tasks/task_test
  ```
- Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream to be queried.
transfer_task_name	Yes	String	Name of the dump task.

Response

- If the operation is successfully performed, status code 204 is returned and the response message body is empty.
- If the operation failed to be performed, identify the failure cause according to the response body and the instructions in **Error Codes**.

Response Code

- Normal

  204 No Content
- Failed

  For more information, see **Error Codes**.

5.20 Querying Dump Task Details

Function

This API is used to query dump task details.

When querying dump task details, you need to specify the stream name and dump task name.

URI

- URI format

  GET /v2/{project_id}/streams/{stream_name}/transfer-tasks/{transfer_task_name}

- Parameter description

  None

Request

- Example request

```
GET https://{endpoint}/v2/{project_id}/streams/stream_test/transfer-tasks/task_test
```

- Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream to be queried.
transfer_task_name	Yes	String	Name of the dump task.

Response

- Example response

```
{
   "partitions":[
      {
         "partitionId":"shardId-0000000000",
         "discard":0,
         "state":"RUNNING",
         "last_transfer_timestamp":1534163694103,
         "last_transfer_offset":2139
      }
   ],
   "destination_type": "OBS",
   "stream_name":"feihang",
   "task_name":"feihang",
   "state":"RUNNING",
   "create_time":1534142793097,
   "last_transfer_timestamp":1534163694103,
   "obs_destination_description":{
      "agency_name":"a",
      "file_prefix":"feihang",
      "partition_format":"",
      "obs_bucket_path":"0feihang",
      "deliver_time_interval":30,
      "consumer_strategy":"LATEST",
      "destination_file_type":"text"
   }
}
```

- Parameter description

Table 5-52 Response parameter description

Parameter	Type	Description
stream_name	String	Name of the DIS stream.
task_name	String	Name of the dump task.

Parameter	Type	Description
create_time	Long	Task creation timestamp.
last_transfer_timestamp	Long	Timestamp of the latest dump.
state	String	Dump task status. Possible values: • **ABNORMAL** • **RUNNING**
destination_type	String	Dump destination. Possible values: • **OBS** • **MRS** • **DLI** • **CLOUDTABLE** • **DWS**
partitions	List<PartitionTask>	List of partition dump details.
obs_destination_description	Object	Parameter list of the OBS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to OBS.
mrs_destination_description	Object	Parameter list of the MRS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to MRS.

Parameter	Type	Description
dli_destination_description	Object	Parameter list of the DLI to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to DLI.
cloudtable_destination_descripton	Object	Parameter list of the CloudTable to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to CloudTable HBase or OpenTSDB.
dws_destination_descripton	Object	Parameter list of the DWS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to DWS.

Table 5-53 PartitionTask parameter description

Parameter	Type	Description
partitionId	String	Unique identifier of the partition.

Parameter	Type	Description
discard	Integer	Total number of dirty data records in the current partition.
state	String	Dump task status. Possible values: • **ABNORMAL** • **RUNNING**
last_transfer_timestamp	Long	Timestamp of the latest dump.
last_transfer_offset	Long	Offset of the latest dump.

Response Code

- Normal
  200 OK
- Failed
  For more information, see **Error Codes**.

5.21 Querying a Dump Task List

Function

This API is used to query a dump task list.

When querying a dump task list, you need to specify the stream name.

URI

- URI format
  GET /v2/{project_id}/streams/{stream_name}/transfer-tasks
- Parameter description
  None

Request

- Example request
  ```
  GET https://{endpoint}/v2/{project_id}/streams/stream-test/transfer-tasks
  ```
- Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream to be queried.

Response

- Example response

```
{
   "tasks":[
      {
         "task_name":"task",
         "create_time":1534142793097,
         "last_transfer_timestamp":1534163694103,
         "state":"RUNNING",
         "destination_type": "OBS"
      },
      {
         "task_name":"another_task",
         "create_time":1534163306123,
         "last_transfer_timestamp":1534163680563,
         "state":"RUNNING",
         "destination_type": "DLI"
      }
   ],
   "total_number":2
}
```

- Parameter description

Table 5-54 Response parameter description

Parameter	Type	Description
total_number	Integer	Total number of dump tasks.
tasks	List<Task>	List of dump tasks.

Table 5-55 Task parameter description

Parameter	Type	Description
task_name	String	Name of the dump task.
create_time	Long	Task creation timestamp.
last_transfer_timestamp	Long	Timestamp of the latest dump.

Parameter	Type	Description
state	Integer	Dump task status. Possible values: • **ABNORMAL** • **RUNNING**
destination_type	String	Dump destination. Possible values: • **OBS** • **MRS** • **DLI** • **CLOUDTABLE** • **DWS**

Response Code

- Normal

  200 OK
- Failed

  For more information, see **Error Codes**.

5.22 Obtaining Stream Consumption Information

Function

This API is used to obtain stream consumption information.

When obtaining the stream consumption information, you need to specify the stream name and application name.

URI

- URI format

  GET /v2/{project_id}/apps/{app}/streams/{stream_name}
- Parameter description

  None

Request

- Example request

  ```
  GET https://{endpoint}/v2/{project_id}/apps/app_name/streams/stream_name?checkpoint_type=LAST_READ&start_partition_id=0&limit=3
  ```

  – Parameter description

Table 5-56 Parameter description

Parameter	Mandatory	Type	Description
limit	No	Int	The number of partitions whose consumption states will be returned. Value range: 1 to 100 Default: **100**
start_partition_id	No	String	Start partition ID for obtaining flow consumption information. Value range: 0 to (the total number of partitions – 1) Default: **0**
checkpoint_type	Yes	String	Type of the checkpoint. Set this parameter to **LAST_READ**.
app	Yes	String	Name of the application.
stream_name	Yes	String	Name of the stream created on the management console. A stream name is 1 to 64 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Response

– Example response

```
{
"stream_name":"${name}",
"app_name":"5395527",
"partition_consuming_states":[
{
"partition_id":"2",
"sequence_number":"-1",
"latest_offset":"0",
"earliest_offset":"0",
"checkpoint_type":"LAST_READ"}

]
}
```

– Parameter description

Table 5-57 Response parameter description

Parameter	Type	Description
stream_name	String	Name of the DIS stream.
app_name	String	Name of the application.
partition_consuming_states	List<partition_consuming_state>	Consumption state of each partition.
has_more	Boolean	Whether consumption states of other partitions are available.

Table 5-58 Partition_consuming_state parameter description

Parameter	Type	Description
partition_id	String	Unique identifier of the partition.
sequence_number	String	Sequence number. The sequence number is defined in the API request in **Adding a Checkpoint**. **NOTE** If the checkpoint is not submitted or has been deleted, the value is **-1**.
latest_offset	Long	Latest offset of the partition data.
earliest_offset	Long	Earliest offset of the partition data.
checkpoint_type	String	Type of the checkpoint. Set this parameter to **LAST_READ**.
metadata	String	Metadata of the checkpoint. For details, see **Adding a Checkpoint**.

Response Code

- Normal

  200 OK

- Failed

  For more information, see **Error Codes**.

5.23 Creating a Stream That Has an OBS Dump Task (Discarded)

NOTE

In earlier versions, you can add an OBS periodic dump task when creating a stream. This function has been discarded in new versions. For details about how to create a stream and add dump tasks for it, see **Creating a DIS Stream** and **Adding a Dump Task**.

Function

This API is used to create a stream.

URI

- URI format

  POST /v2/{project_id}/streams

- Parameter description

  None

Request

- Example request

```
POST https://{endpoint}/v2/{project_id}/streams

{
  "stream_name": "obs_test",
  "partition_count": 1,
  "obs_destination_descriptor": [
    {
      "agency_name": "agency",
      "obs_bucket_path": "1111",
      "file_prefix": "",
      "deliver_time_interval":300,
      "partition_format": "yyyy/MM/dd/HH/mm",
      "record_delimiter": "\n"


    }
  ]
}
```

- Parameter description

Table 5-59 Parameter description

Parameter	Mandatory	Type	Description
stream_name	Yes	String	Name of the DIS stream. Name of the DIS stream to be created. A stream name is 1 to 64 characters long. Only letters, digits, hyphens (-), and underscores (_) are allowed.
stream_type	No	String	Partition type. Possible values: • **COMMON**: a common stream. The bandwidth is 1 MB/s. • **ADVANCED**: an advanced stream. The bandwidth is 5 MB/s. Default value: COMMON
partition_count	Yes	Int	Number of the partitions. Partitions are the base throughput unit of a DIS stream. The value range varies depending on the value of **stream_type**. • If **stream_type** is set to **COMMON**, the value of **partition_count** is an integer from 1 to 50. If a tenant has created *N* common partitions, then the tenant can create **50-N** more common partitions. • If **stream_type** is set to **ADVANCED**, the value of **partition_count** is an integer from 1 to 10. If a tenant has created *N* advanced partitions, then the tenant can create **10-N** more advanced partitions.
data_duration	No	Int	Period of time for which data is retained in the DIS stream. Value range: *N* x 24, where *N* is an integer from 1 to 7. Unit: hour Default value: 24 If this parameter is left unspecified, the default value will be used.

Parameter	Mandatory	Type	Description
data_type	No	String	Source data type. Possible values: • **BLOB**: binary data. Data of BLOB type can be stored in DIS or dumped to OBS, MRS, DWS, or DLI. • **JSON**: data with a strong schema structure. Data of JSON type can be stored in DIS or dumped to CloudTable. Default value: **BLOB**
obs_destination_descriptor	No	Object	Parameter list of the OBS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to OBS.
mrs_destination_descriptor	No	Object	Parameter list of the MRS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to MRS.
dli_destination_descriptor	No	Object	Parameter list of the DLI to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to DLI.

Parameter	Mandatory	Type	Description
cloudtable_destination_descriptor	No	Object	Parameter list of the CloudTable to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to CloudTable HBase or OpenTSDB.
dws_destination_descriptor	No	Object	Parameter list of the DWS to which data in the DIS stream will be dumped. Data in a DIS stream cannot be dumped to multiple destinations. This parameter is left unspecified by default. If this parameter is left unspecified, data is not dumped to DWS.

Table 5-60 obs_destination_descriptor parameter description

Parameter	Mandatory	Type	Description
agency_name	Yes	String	Name of the agency created in IAM. DIS uses an agency to access your specified resources. Agency parameter settings: • **Agency Type**: **Cloud service** • **Cloud Service**: **DIS** • **Validity Period**: **Permanent** • Region: Global service Project: OBS Policy: Tenant Administrator This parameter cannot be left unspecified and the parameter value cannot exceed 64 characters.
obs_bucket_path	Yes	String	Name of the OBS bucket used to store data from the DIS stream.

Parameter	Mandatory	Type	Description
file_prefix	No	String	Directory to store files that will be dumped to OBS. Different directory levels are separated by slashes (/) and cannot start with slashes. **NOTE** • This parameter is optional if streaming data is dumped to OBS. • This parameter is not required if small files are dumped to OBS. A directory name cannot exceed 50 characters. Only letters, digits, and underscores (_) are allowed. The priority order is as follows: • **file_prefix**: high • **partition_format**: medium • **DIS_generated_filename**: low
deliver_time_interval	No	Int	User-defined interval at which data is imported from the current DIS stream into OBS. If no data is pushed to the DIS stream during the current interval, no dump file package will be generated. **NOTE** • This parameter is mandatory if streaming data is dumped to OBS. • This parameter is not required if small files are dumped to OBS. Value range: 30s to 900s Default value: 300 Unit: second

Parameter	Mandatory	Type	Description
partition_format	No	String	Directory structure of the object file written into OBS. The directory structure is in the format of yyyy/MM/dd/HH/mm (time at which the dump task was created). • YYYY: year • MM: month • dd: day • HH: hour • mm: minute For example, if the dump task was created at 14:49:00 on November 10, 2017, then the directory structure is **2017/11/10/14/49**. **2017** is the level-1 directory.
deliver_data_type	No	String	This parameter is mandatory if **Dump Type** is set to **Custom file**. Default value: file_stream
record_delimiter	No	String	Delimiter for the dump file, which is used to separate the user data that is written into the dump file. Possible values: • Comma (,) • Semicolon (;) • Vertical bar (|) • Newline (\n) • NULL Default value: newline (\n)

NOTE

- The data dumped to OBS is the raw data uploaded by users, that is, the data decoded using Base64.
- The records uploaded by users are separated by newline (\n).

Response

- If the DIS stream was successfully created, a 201 response with an empty response body is returned. For details about status codes, see **Status Codes**.
- If the DIS stream failed to be created, identify the failure cause according to the response body and the instructions in **Error Codes**.

Response Code

- Normal
  201 Created
- Failed
  For more information, see **Error Codes**.

6 Appendix

6.1 Error Codes

If API calling fails, a response with HTTP status code 4*xx* or 5*xx* is returned. The returned message body contains the specific error code and error information. The response also includes specific error code and error information in the message body, as described in the following table.

Sample:

{

"errorCode": "DIS.4301",

"message": "Stream does not exist. [test][6332998f84ac4c13a83db055da33cb66]"

}

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**.

HTTP Status Code	Error Code	Error Message	Description	Measure
441	DIS.4100	Authorization error.	The signature information generated using AK and SK is incorrect.	Ensure that the signature information in the request header is correct.

HTTP Status Code	Error Code	Error Message	Description	Measure
441	DIS.4101	Authorization header cannot be empty.	The signature information generated using AK and SK is blank.	Ensure that the signature information is generated.
441	DIS.4102	Incorrectly parsed authorization header.	The signature cannot be parsed.	Ensure that the signature information in the request header is correct.
441	DIS.4103	Empty X-Sdk-Date header.	The **X-Sdk-Date** field in the request header is blank.	Ensure that the **X-Sdk-Date** field in the request header is not blank.
441	DIS.4104	Error parsing X-Sdk-Date header.	The **X-Sdk-Date** field in the request header cannot be parsed.	Ensure that the **X-Sdk-Date** field in the request header is correct.
441	DIS.4105	Invalid X-Sdk-Date header.	The **X-Sdk-Date** field in the request header is invalid.	Ensure that the **X-Sdk-Date** field in the request header is correct.

HTTP Status Code	Error Code	Error Message	Description	Measure
441	DIS.4106	Empty AcessKey header.	The **Authorization** field of the request header does not contain AK.	Ensure that AK is contained in the **Authorization** field.
441	DIS.4107	Invalid AcessKey header.	AK in the **Authorization** field of the request header is invalid.	Ensure that AK is valid and correct.
441	DIS.4108	Empty ServiceName header.	The **Authorization** field of the request header does not contain the service name.	Ensure that the **Authorization** field of the request header contain service name **dis**.
441	DIS.4109	The Authorization header must contain the following field: {Credential,SignedHeaders,Signature;}	The **Authorization** field of the request header is incorrect.	Ensure that the **Authorization** field of the request header contains **Credential**, **SignedHeaders**, and **Signature**.

HTTP Status Code	Error Code	Error Message	Description	Measure
441	DIS.4110	Empty Signature header.	The **Authorization** field of the request header does not contain **SignedHeaders**.	Ensure that the signature generation mode is correct.
441	DIS.4111	Invalid Region header.	The region in the **Authorization** field of the request header is invalid.	Ensure that the region is valid.
441	DIS.4112	Invalid authorization request.	The signature information generated using AK and SK is incorrect.	Ensure that the signature generation mode and the information about AK, SK, and region are correct.
441	DIS.4113	Empty Token header.	When token authentication is used, the **X-Auth-Token** field of the request header is blank.	Ensure that the **X-Auth-Token** field of the request header is not blank.

HTTP Status Code	Error Code	Error Message	Description	Measure
441	DIS.4114	Invalid Token header.	When token authentication is used, the **X-Auth-Token** field of the request header is invalid.	Ensure that the **X-Auth-Token** field of the request header is valid.
403	DIS.4116	Invalid RBAC.	User operations are restricted.	Ensure that real-name authentication have been performed, all bills have been paid, and DIS operating permissions have been obtained.
400	DIS.4117	Invalid Project Id.	The project ID input by the subscriber is invalid.	Ensure that the project ID is valid and correct.
400	DIS.4200	Invalid request.	The user request is invalid.	Check the request by referring to the API document.
400	DIS.4201	Invalid partition_id.	The partition ID input by the subscriber is invalid.	Ensure that the partition ID is valid.
400	DIS.4202	Empty request.	The user request is empty.	Input a valid request.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4203	Invalid monitoring period.	The start time for query the monitoring information is invalid.	Input a valid timestamp.
400	DIS.4204	The monitoring period cannot be longer than 7 days.	Only the monitoring information generated in the recent seven days can be queried.	Query the monitoring information generated in the recent seven days.
400	DIS.4205	Stream is not running.	The stream is not in the running state.	Check the stream status.
400	DIS.4208	Invalid MRS cluster.	The MRS cluster input during MRS dump task creation is invalid.	Ensure the MRS cluster name and ID are correct and the cluster is running in security mode.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4209	Invalid metrics label.	The monitoring metric input during monitoring information query is invalid.	Check and correct the monitoring metric by referring to the API document.
400	DIS.4215	Invalid cursor type.	The cursor type input during the data cursor acquisition is invalid.	Check and correct the cursor type by referring to the API document.
400	DIS.4216	Invalid sequence_number.	The sequence number input during data cursor acquisition is invalid.	Input a valid sequence number.
400	DIS.4217	Invalid partition cursor.	The partition cursor input during data download from DIS is invalid.	Obtain the partition cursor again and download the data.
400	DIS.4219	The file is constantly resent.	The file has been received.	Do not upload the file again.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4220	The block whose sequence number is %s needs to be resent.	The file block needs to be uploaded again.	Upload the corresponding file block as instructed.
400	DIS.4221	Block seq %s is expected.	Duplicate file blocks are input.	Upload the file block from the one expected by the system.
400	DIS.4222	Block seq %s is expected.	The input file blocks are not consecutive.	Upload the file block from the one expected by the system.
400	DIS.4223	The file size exceeds the limit.	The file capacity exceeds the upper limit.	Split the file and upload it again.
400	DIS.4224	The sequence number is out of range.	The sequence number input during data cursor acquisition is invalid.	Input a valid sequence number.
400	DIS.4225	Expired partition cursor.	The partition cursor input during data download from DIS expires.	Obtain the partition cursor again and download the data.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4226	A partition iterator error occurred or a record to which the SN corresponds has expired. Try to obtain the partition iterator again.	The sequence number of the partition cursor input during data acquisition expires.	Obtain the data cursor and use the new cursor to obtain data.
400	DIS.4300	Request error.	The request body error occurs.	Correct the request body by referring to the API document.
400	DIS.4301	The stream does not exist.	The stream does not exist.	Ensure that the stream exists.
400	DIS.4302	The partition does not exist.	The stream partition does not exist.	Ensure that the partition ID exists.
400	DIS.4303	The traffic control limit is exceeded.	The flow control limit is exceeded.	Add partitions or reduce the upload rate.
400	DIS.4305	Too many stream requests.	There is an excessive number of user requests at the same time.	Decrease the requesting frequency and try again.
400	DIS.4306	The bucket does not exist.	The OBS bucket does not exist.	Ensure that the OBS bucket exists.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4307	The stream already exists.	The stream already exists.	Change the name of the new stream.
400	DIS.4308	Insufficient quota.	The quotas of the streams or partitions are insufficient.	Release the quota or **submit a service ticket** to change the quota of the account.
400	DIS.4309	Too many request failures. Please try again later.	The IP address is added to the blacklist.	Ensure that the authentication information and request are valid and try again later.
400	DIS.4310	OBS access error.	OBS failed to be accessed.	Ensure that the user has permission to access OBS.
400	DIS.4319	Partition is expired.	The partition has expired.	The partition has expired. Check whether the partition has expired in the scaling-in scenario and use a correct and valid partition.
400	DIS.4329	app quota exceeded.	The application quota exceeds the limit.	Release the applications that are not used.
400	DIS.4330	app already exist.	An application with the same name already exists.	Change the name of the new application.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4331	app is using.	The application failed to be deleted.	Ensure that the application that you want to delete is not being used.
400	DIS.4332	app not found.	The specified application does not exist.	Ensure the application name is correct.
400	DIS.4335	Invalid IAM agency.	The IAM agency used during dump task creation is invalid.	Ensure that **dis_admin_agency** created by DIS or the user-defined IAM agency exists and permission is complete.
400	DIS.4336	Invalid HDFS path.	The MRS HDFS path input during MRS dump task creation is invalid.	Ensure that the MRS HDFS path exists.
400	DIS.4337	The DLI database does not exist.	The DLI database input during DLI dump task creation does not exist.	Ensure that the DLI database exists.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4338	The DLI table does not exist.	The DLI table input during DLI dump task creation does not exist.	Ensure that the DLI table exists and is an internal table.
400	DIS.4339	Consumer quota exceeded.	The consumer quota of the consumer group is insufficient.	The number of consumers in the consumer group has exceeded the maximum quota. Allocate consumers properly or create a consumer group to meet the requirement.
400	DIS.4341	The CloudTable cluster does not exist.	The CloudTable cluster input during CloudTable dump task creation does not exist.	Ensure that the CloudTable cluster exists and is running properly.
400	DIS.4342	The CloudTable table does not exist	The CloudTable table input during CloudTable dump task creation does not exist.	Ensure that the CloudTable table exists.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4343	The CloudTable table family does not exist.	The CloudTable column family input during CloudTable dump task creation does not exist.	Ensure that the CloudTable column family exists.
400	DIS.4345	Invalid CloudTable schema.	The schema input during CloudTable dump task creation is invalid.	Check the schema based on the returned details to ensure that the configured JSON attribute name exists and the parameters are valid.
400	DIS.4348	Invalid CloudTable openTSDB schema.	The schema input during CloudTable OpenTSDB dump task creation is invalid.	Check the schema based on the returned details to ensure that the JSON attribute name exists and the parameters are valid.
400	DIS.4350	Invalid DWS cluster.	The DWS cluster input during DWS dump task creation does not exist.	Ensure that the DWS cluster exists and is running properly.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4351	Invalid KMS userKey.	The KMS key input during DWS dump task creation is invalid.	Ensure that the KMS key exists.
400	DIS.4354	The transfer task does not exist.	The dump task to be deleted or updated does not exist.	Ensure that the dump task exists.
400	DIS.4355	The transfer task already exists.	When you create a dump task in a stream, another dump task with the same name already exists in this stream.	Change the name of the new dump task.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4357	Exceeded transfer task quota.	A maximum of five dump tasks can be created for one stream at the same time. Creating six dump tasks will exceed the quota limit.	Delete the discarded dump tasks and then add dump tasks again.
400	DIS.4358	The stream supports specific transfer tasks. Check the data type of the stream.	Common dump tasks cannot be created in the stream for small file dump.	Create a new stream and then dump tasks in the stream.
400	DIS.4360	Invalid data schema.	The data schema input during stream creation or update is invalid.	Ensure that the data schema format is correct and try again.
400	DIS.4601	The number of resource tags has reached the maximum.	A resource has a maximum of 10 tags. Adding 11 tags will exceed the quota limit.	Delete the discarded tags and then add tags again.

HTTP Status Code	Error Code	Error Message	Description	Measure
400	DIS.4602	Invalid resource type.	The resource type is invalid.	Ensure that the resource type is valid.
400	DIS.4603	The resource does not exist.	The resource does not exist.	Ensure that the resource is not deleted.
400	DIS.4604	The key does not exist.	The tag key does not exist.	Ensure that the tag key exists.
400	DIS.4605	The action is not supported.	The current tag operation is not supported.	Ensure that the current tag operation is valid. Currently, only the create and delete operations are supported.
400	DIS.4375	The app does not commit checkpoint	The application does not submit the checkpoint operation in the stream.	Check whether the application has submitted the checkpoint operation in the consumption stream.
500	DIS.5000 to DIS.5999	System error. **NOTE** Contact technical support to handle system errors.	-	-

6.2 Status Codes

A status code is an HTTPS response issued by DIS to indicate whether an API request has been successfully completed.

Status Code	Status	Description
100	Continue	The server has received the initial part of the request and the client should continue to send the remaining part. It is issued on a provisional basis while request processing continues. It alerts the client to wait for a final response.
101	Switching Protocols	The requester has asked the server to switch protocols and the server has agreed to do so. The target protocol must be more advanced than the source protocol. For example, the current HTTP protocol is switched to a later version of HTTP.
200	OK	The server has successfully processed the request.
201	Created	The request has been fulfilled, resulting in the creation of a new resource.
202	Accepted	The request has been accepted for processing, but the processing has not been completed.
203	Non-Authoritative Information	The server successfully processed the request, but is returning information that may be from another source.
204	NoContent	The server has successfully processed the request, but does not return any content. The status code is returned in response to an HTTPS OPTIONS request.
205	Reset Content	The server has successfully processed the request, but does not return any content. Unlike a 204 response, this response requires that the requester reset the content.
206	Partial Content	The server has successfully processed a part of the GET request.
300	Multiple Choices	There are multiple options for the requested resource. For example, this code could be used to present a list of resource characteristics and addresses from which the client such as a browser may choose.
301	Moved Permanently	This and all future requests should be permanently directed to the given URI indicated in this response.
302	Found	The requested resource was temporarily moved.

Status Code	Status	Description
303	See Other	The response to the request can be found under another URI using a GET or POST method.
304	Not Modified	The requested resource has not been modified. In such a case, there is no need to retransmit the resource since the client still has a previously-downloaded copy.
305	Use Proxy	The requested resource is available only through a proxy.
306	Unused	This HTTP status code is no longer used.
400	BadRequest	The request is invalid. The client should modify the request instead of re-initiating it.
401	Unauthorized	The authentication information provided by the client is incorrect or invalid.
402	Payment Required	Reserved for future use.
403	Forbidden	The server has received the request and understood it, but the server is refusing to respond to it. The client should modify the request instead of re-initiating it.
404	NotFound	The requested resource could not be found. The client should modify the request instead of re-initiating it.
405	MethodNotAllowed	A request method is not supported for the requested resource. The client should modify the request instead of re-initiating it.
406	Not Acceptable	The server could not fulfill the request according to the content characteristics of the request.
407	Proxy Authentication Required	This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Time-out	The server timed out waiting for the request. The client may re-initiate the request without modifications at any later time.

Status Code	Status	Description
409	Conflict	The request could not be processed due to a conflict in the request. For example, an edit conflict between multiple simultaneous updates or the resource that the client attempts to create already exists.
410	Gone	The requested resource has been deleted permanently and will not be available again. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server refused to process the request because the request does not specify the length of its content.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.
413	Request Entity Too Large	The request is larger than the server is willing or able to process. The server may close the connection to prevent the client from continuing the request. If the server temporarily cannot process the request, the response will contain a Retry-After header field.
414	Request-URI Too Large	The URI provided was too long for the server to process.
415	Unsupported Media Type	The server does not support the media type in the request.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the Expect request-header field.
422	UnprocessableEntity	The request was well-formed but was unable to be followed due to semantic errors.
429	TooManyRequests	The client has sent more requests than its rate limit is allowed within a given amount of time, or the server has received more requests than it is able to process within a given amount of time. In this case, it is advisable for the client to re-initiate requests after the time specified in the Retry-After header of the response expires.

Status Code	Status	Description
441	Authentication Error	Authentication fails.
500	InternalServerError	The server is able to receive the request but it could not understand the request.
501	Not Implemented	The server does not support the requested function.
502	Bad Gateway	The server was acting as a gateway or proxy and received an invalid request from a remote server.
503	ServiceUnavailable	The requested service is invalid. The client should modify the request instead of re-initiating it.
504	ServerTimeout	The server could not return a timely response. The response will reach the client only if the request carries a timeout parameter.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

6.3 Obtaining a Project ID

This section describes how to obtain a project ID on the console or by calling an API.

Obtaining a Project ID by Calling an API

You can obtain a project ID by calling the API for **Querying Project Information Based on the Specified Criteria**.

The API for obtaining the project ID is **GET https://{Endpoint}/v3/projects/**, where *Endpoint* indicates the endpoint of IAM. You can obtain the endpoint from **Regions and Endpoints**. For details about API authentication, see **Authentication**.

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

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



```
            },
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

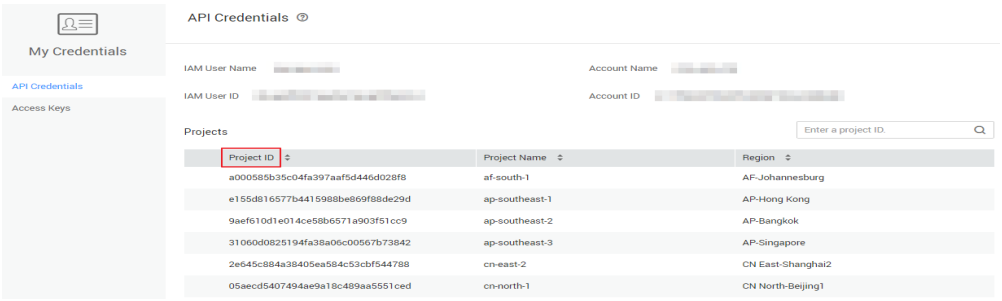
Obtaining a Project ID from the Console

A project ID is required for some URLs when an API is called. To obtain a project ID, perform the following operations:

1. Log in to the management console.
2. Hover the mouse over the username and select **Basic Information**.
3. On the displayed page, click **Manage** in **Security Credentials**.

   On the **API Credentials** page, view project IDs in the project list.

Figure 6-1 Viewing project IDs



A Change History

Release Date	What's New
2020-07-03	This is the thirty-second official release. Added error codes. For details, see **Error Codes**.
2020-05-09	This is the thirty-first official release. Allowed users to view monitoring information of applications. For details, see **Downloading Data from a DIS Stream**.
2019-12-24	This issue is the thirtieth official release. Added the description of obtaining a token. For details, see **Authentication**. Modified the URIs in all API example requests and deleted parameter **port**.
2019-12-02	This issue is the twenty-ninth official release. Modified the data retention period. For details, see **Creating a DIS Stream**.
2019-10-08	This issue is the twenty-eighth official release. Added the description of the pagination query function. For details, see **Listing DIS Streams**.
2019-09-30	This issue is the twenty-seventh official release. Added the error tolerance parameter to the DWS dump API. For details, see **Adding a Dump Task**.
2019-09-11	This issue is the twenty-sixth official release. Added the function of calling the API to obtain the project ID in **Obtaining a Project ID**.
2019-08-31	This issue is the twenty-fifth official release. Hid cn-north-1 in **Making an API Request**.

Release Date	What's New
2019-07-23	This issue is the twenty-fourth official release. Modified the JSON format. For details, see **Downloading Data from a DIS Stream** and **Adding a Dump Task**.
2019-05-14	This issue is the twenty-third official release. Modified the following section: **Viewing Details of a DIS Stream**
2019-01-07	This issue is the twenty-second official release. Modified the following section: • **Adding a Dump Task**